Requested Patent: EP0465090A1

Title:

CONGESTION CONTROL FOR CONNECTIONLESS TRAFFIC IN DATA NETWORKS VIA ALTERNATE ROUTING. ;

Abstracted Patent: EP0465090 ;

Publication Date: 1992-01-08 ;

Inventor(s): DRAVIDA SUBRAHMANYAM (US); HARSHAVARDHANA P (US) ;

Applicant(s): AMERICAN TELEPHONE _TELEGRAPH (US) ;

Application Number: EP19910305728 19910625 ;

Priority Number(s): US19900548457 19900703 ;

IPC Classification: H04L12/56 ;

Equivalents: DE69118974D, DE69118974T, JP2679895B2, JP4233848, US5253248

ABSTRACT:

A congestion control scheme for connectionless networks relieves congestion by routing a portion of traffic on a congested primary path onto a predefined alternate path constructed such that loop-freedom is guaranteed. Explicit care is taken to avoid spreading congestion onto alternate paths. The control actions are taken in a completely distributed manner, based on local measurements only and therefore no signaling messages need to be exchanged between nodes. If desired, lower loss priority may be assigned to alternate routed traffic. Congestion is monitored locally and thresholds defined to declare the onset and abatement of congestion. The present invention affords at least an order of magnitude improvement in end-to-end cell blocking under sustained focussed overload.

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number : **0 465 090 A1**

(12) **EUROPEAN PATENT APPLICATION**

(21) Application number : **91305728.7**

(51) Int. Cl.^{5} : **H04L 12/56**

(22) Date of filing : **25.06.91**

(30) Priority : **03.07.90 US 548457**

(43) Date of publication of application :
08.01.92 Bulletin 92/02

(84) Designated Contracting States :
DE FR GB IT

(71) Applicant : **AMERICAN TELEPHONE AND TELEGRAPH COMPANY**
550 Madison Avenue
New York, NY 10022 (US)

(72) Inventor : **Dravida, Subrahmanyam**
507 Santa Anita Lane
Toms River, New Jersey 08755 (US)
Inventor : **Harshavardhana, P.**
2 Birch Hill Road
Freehold New Jersey 07725 (US)

(74) Representative : **Buckley, Christopher Simon Thirsk et al**
AT&T (UK) LTD. 5 Mornington Road
Woodford Green, Essex IG8 OTU (GB)

(54) **Congestion control for connectionless traffic in data networks via alternate routing.**

(57) A congestion control scheme for connectionless networks relieves congestion by routing a portion of traffic on a congested primary path onto a predefined alternate path constructed such that loop-freedom is guaranteed. Explicit care is taken to avoid spreading congestion onto alternate paths. The control actions are taken in a completely distributed manner, based on local measurements only and therefore no signaling messages need to be exchanged between nodes.

If desired, lower loss priority may be assigned to alternate routed traffic. Congestion is monitored locally and thresholds defined to declare the onset and abatement of congestion. The present invention affords at least an order of magnitude improvement in end-to-end cell blocking under sustained focussed overload.

EP 0 465 090 A1

Technical Field

The present invention relates generally to data communications, and, in particular, to a congestion control scheme for connectionless traffic in data networks.

Background of the Invention

Connectionless data networks (such as the ARPANET network) permit the interchange of packetized data between interconnected nodes without the need for fixed or centralized network routing administration. Each node examines packet header information and makes routing decisions based only upon locally available information, without explicit knowledge of where the packet originated or of the entire route to the destination node. In this environment, traditional congestion control strategies such as window flow control and per virtual circuit buffering and pacing cannot be used because of the absence of end-to-end acknowledgements.

One congestion control approach that has been implemented in some connectionless networks is the use of choke messages. In this method, a congested node sends feedback messages to other nodes, asking them not send traffic to it until further notification. There are several drawbacks to this approach: first, by the time the choke message reaches the offending node, a substantial amount of traffic would have been transmitted. For example, in a network consisting of 150 Mbps trunks, a choke packet sent on 1000 mile long link takes 10 msecs of propagation time. In this time, 1.5 M bits are already in transit and will contribute to existing congestion. Secondly, in connectionless networks, there is no knowledge of the path traversed by a packet before arriving at a given node; therefore, choke messages may have to be sent to all the neighbors including those that do not contribute to congestion. This will lead to under-utilization of the network. Another difficulty with this method is the action taken by a node upon receiving a choke packet. If it drops all packets headed towards the congested node, then subsequent retransmissions will contribute to increased congestion. Since there is no connection-oriented layer that the network interacts with, it is difficult to stop traffic at the sources responsible for causing congestion. Therefore choke messages do not appear to be an effective means of congestion control in connectionless networks.

Certain other approaches that have been tried in connectionless networks such as ARPANET involve changing network routing in response to changes in traffic conditions, by dynamically recomputing paths between nodes in a completely distributed fashion. This can be illustrated by considering the RIP scheme which has been tried in ARPANET. In RIP, each node stores the entire network topology, and periodically transmits routing update messages to its neighboring nodes. The routing update messages provide reachability information which tells each neighboring node how the originating node can reach the other nodes in the network, together with some measure of the minimum distance to the various nodes. The measure of distance used is different in different versions of RIP. The original RIP protocol used hop-counts to measure distance, while subsequent modifications use delay estimates to reach a destination as a measure of distance.

The problem with the RIP scheme is that it has several serious drawbacks: first, a large amount of information must be exchanged between nodes in order to ensure consistent routing changes, and this itself may consume significant network resources. Second, because paths are dynamically recomputed, there is serious potential for problems such as packet looping, packet missequencing and route oscillations. Also, because of propagation delay, the information exchanged between nodes may be outdated, and hence may not be reliable for changing routing. This problem is especially serious in high speed networks (> 45 mbps).

A second dynamic routing protocol called IGRP uses a composite metric which includes propagation delay, path bandwidth, path utilization and path reliability, as a measure of distance. If the minimum distance path is different from the one currently in use, then all the traffic is switched to the newly computed shortest path. If a set of paths are "equivalent", load balancing is used.

When dynamic changes in routing are occasioned by the IGRP protocol, traffic shifts from one path to another, so that congestion may be caused on the new path. Subsequent distance and shortest path computation may then switch the traffic back onto the original path. In this manner each path would experience oscillations in offered traffic and the end result may well be that neither path is fully utilized. This problem may only be partly alleviated by averaging the distance measurements over an interval of time before transmitting to the other nodes.

A third, very recent proposed enhancement to the ARPANET routing protocol decribed in "An Extended Least-Hop Distributed Routing Algorithm," written by D. J. Nelson, K. Sayood, and H. Chang, published in IEEE Transactions on Communications, Vol. 38, No. 4, April 1990, pages 520-528, augments the set of available shortest path routes to carry packets to a given destination by including routes that are one hop longer than the shortest path routes. Each node maintains an estimate of the total delay involved in reaching every destination. The route which has the minimum delay to a given destination is then picked from the set of routes avail-

able to carry traffic to that destination. Although this approach shows considerable improvement over the existing ARPANET routing, it also has several disadvantages. First, the optimal, minimum delay path has to be chosen for each packet, leading to increased processing in the switch. Second, at any given time, only one path is active and hence there is no notion of load balancing. All traffic is routed on the same path until a path with a better delay estimate is available. Third, nodes need to exchange delay information and hence some form of signaling between nodes is necessary. Lastly, only paths that are one hop longer are considered in addition to the shortest paths. Thus, some longer idle paths will not be chosen, even though they could have successfully carried the traffic.

Yet another possibility for dealing with congestion is to try to reduce the impact of its consequences. For example, one way of avoiding packet losses due to buffer overflow is to increase the link buffer sizes. There is a serious drawback to this approach: if the buffer size is made very large, cells will experience high queueing delays and end-to-end performance may be affected to the extent that the end systems may time-out and retransmit. On the other hand, if the buffer size is designed to keep the maximum queueing delay within acceptable bounds, then since the buffer occupancy tends to increase exponentially as the link utilization approaches unity, buffers will eventually overflow in the face of sustained focussed overload on the link and the resulting cell losses will cause the end systems to retransmit. Thus, increasing the buffer size is not a viable congestion control strategy.

Summary of the Invention

In accordance with the present invention, congestion caused by transient focussed overloads in connectionless networks is relieved by routing a portion of traffic intended for a congested primary path onto a predefined alternate path. An explicit algorithm is used for constructing alternate paths in such a way that loop-freedom is guaranteed. Briefly, this is done by organizing the nodes that neighbor a given node into layers such that nodes that are the same distance (in hops) from a given destination are in the same layer. A weight is then assigned to each possible path between (a) the given node and each neighbor in the same layer, and (b) each neighbor and a node in a closer layer (in hops) to which the neighboring node is connected. The pairwise sum of the weights for each combination of paths is then computed and the alternate path is determined as the path having the minimum sum. Furthermore, care is taken to avoid spreading congestion onto alternate paths by marking alternately routed packets, so that they are more readily droped in the event that congestion is again encountered at nodes further along the alternate path. By appropriately choosing threshold values for initiating a transition to an alternate route and for revering to a primary route, route oscillations can be avoided. The routing determinations and network control actions are taken in a completely distributed manner based on local measurements only, and therefore no signaling messages or routing data need to be exchanged between network nodes. The invention is most useful in conjunction with data networks where traffic tends to be very bursty, because when some paths are busy, it is quite likely that others are relatively idle. Accordingly, when there is non-coincidence of overloads on various parts of the network, our invention provides the greatest benefits.

Brief Description of the Drawing

The present invention will be more fully appreciated by reference to the following detailed description, when read in light of the accompanying drawing in which:

Fig. 1 is a diagram illustrating the interconnection of an exemplary network having seven nodes;

Figs. 2-4 illustrate the "exclusionary trees" developed by one of the nodes in the network of Fig. 1;

Figs. 5-7 illustrate the "exclusionary trees" received by one of the nodes in the network of Fig. 1;

Figs. 8-10 illustrate the "exclusionary trees" of Figs. 5-7, respectively, which have been redrawn so that each successive node descending from a root node is placed at the same vertical level;

Fig. 11 illustrates the result when the "exclusionary trees" of Figs. 8- 10 are merged;

Fig. 12 illustrates the "layering" of nodes in a network with respect to a destination node;

Fig. 13 illustrates primary and alternate paths between some of the nodes in Fig. 12;

Figs. 14 and 15 illustrate undesirable single link looping between a pair of nodes;

Fig. 16 illustrates one example of our alternate routing technique as applied to a four node network in which three nodes are located in a first layer and the fourth node is located in a second layer;

Fig. 17 is a redrawn version of Fig. 16 in which the fourth node is replaced by three "equivalent" nodes;

Fig. 18 illustrates alternate and primary routing paths between a first node i and a second node j;

Fig. 19 illustrates alternate routing between the three nodes in the first layer of Fig. 17 that would lead to undesirable looping;

Fig. 20 illustrates multiple nodes in a network, and arangement of such nodes in k layers;

Fig. 21 is a flow chart illustrating the overall process for generating alternate routes in accordance with the invention;

Figs. 22 and 23 are a flow chart (in two parts) that illustrates the process for performing layering step 2103 of Fig. 21;

Figs. 24 and 25 are a flow chart (in two parts) that illustrates in more detail the process of generating alternate routes in accordance with the invention; and

Fig. 26 is a flow chart illustrating the process for giving higher priority to packets that are routed on uncongested routes and for allowing marked packets that travel on alternate routes because of congestion to be discarded in the event that heavy traffic is encountered;

Fig. 27 is a typical functional architecture for the nodes in the networks of Figs. 1-20;

Fig. 28 is a functional diagram of the arrangement of nodal processor 2730 of Fig 27;

Fig. 29 is a three node network model used in simulations of the present invention;

Fig. 30 is a queueing model corresponding to the network of Fig. 29;

Fig. 31 is a graph illustrating blocking probability with and without alternate routing as a function of offered load; and

Fig. 32 is a rescaled version of the graph of Fig. 31.

Detailed Description

In order to fully understand the alternate routing technique of the present invention, it is instructive to first consider one technique that can be employed to determine the primary path taken by messages traveling between nodes in a connectionless network under normal (i.e., non-congested) conditions. This technique is distributed adaptive minimum spanning tree routing, sometimes also known as "exclusionary tree" routing, details of which can be found in Patent No. 4,466,060 issued to G. G. Riddle on August 14, 1984. Other routing techniques are described in D. E. Comer's book, "Internetworking With TCP/IP: Principles, Protocols and Architecture," Chapter 15: Interior Gateway Protocols, Prentice Hall, 1988. The overall objective of the exclusionary tree technique is to maintain a table of correct shortest paths to all destinations at each node of the network. For this purpose, routing tables are initially constructed and updated whenever there are topological changes in the network, as for example, when a node or link is added or deleted. The update procedures are implemented at each node independently, in a distributed fashion. The resulting routing tables are designed to yield minimum hop paths to all destinations in such a way that there is no looping.

Two principal steps are at the heart of the exclusionary tree routing technique: (1) Each node sends an exclusionary tree to each of its neighbors, and (2) a prescribed procedure is employed at each of the nodes to merge the received exclusionary trees into a routing table. These two steps are repeated at each node until the routing tables converge.

The exclusionary tree routing technique can be best described through the following example. Consider the network consisting of seven nodes 1-7 shown in Fig. 1. Each node sends an exclusionary tree to each of its neighbors. An exclusionary tree is the shortest path tree obtained after deleting all links connected to the receiving node. Figs. 2-4 illustrate the exclusionary trees sent by node 1 to its neighbors, namely nodes 6, 5 and 2, respectively. Figs. 5-7 show the exclusionary trees received by node 1 from its neighbors, nodes 6, 4 and 2, respectively. The received exclusionary trees are each first redrawn with their nodes descending from the root, each successive node being placed at a vertical level corresponding to its distance in hops from node 1, as shown in Figs. 8-10. The merged tree for node 1 shown in Fig. 11 is obtained by merging the exclusionary trees of Figs. 8- 10 received by node 1 from its neighbors, according to the following procedure: The received exclusionary trees' nodes at a distance of one hop are visited from left to right (in the example, node 6, then node 5, then node 2) and placed in the merged tree of Fig. 11). Next, nodes at a distance of two hops are visited in the same order (left to right) and are attached to their parent nodes, if they are not already there at a lesser distance. This procedure is repeated successively to create a merged tree. If the node of interest is present in more than one received exclusionary tree at the same distance in hops, then each root node is represented in the merged tree, resulting in multiple entries for nodes that have multiple equal length routes. This situation did not occur in Fig. 11. Whenever multiple equal length routes exist, traffic is distributed over all such routes so as to achieve load balancing. It is to be noted here that other techniques can also be used to determine the primary network routing used in the absence of congestion.

In accordance with our invention, during times of congestion, some fraction of the packets normally routed on primary routes are instead routed on secondary or alternate paths that are lightly loaded. The manner in which alternate routes are selected will be better understood by first considering an arbitrary network which is depicted in the form of a layered architecture in Fig. 12. The layering in Fig. 12 is with respect to destination node D, such that nodes 1231-1233 in layer k (k is an integer) have at least one k-hop shortest path to D. This

means that every node in layer k must have at least one link connecting it to a node in layer (k-l). If a node in layer k is connected to more than one node (1221-1223) in layer (k-l), then it has more than one k-hop shortest path to D. These are precisely the multiple shortest paths constructed by the exclusionary tree routing algorithm described above. By exploiting the connectivity between nodes 1231 - 1233 in layer k, our technique is used to generate loop-free alternate paths to D which are at least of length (k+l) hops. There are two ways of doing this. Both assume that only shortest path primary routes are permitted.

In the first method, if all nodes are numbered, and if we let nodes i and j (i and j are integers) belong to layer k and let nodes i and j be connected by a link, then node i can alternate route packets intended for destination D via node j if i<j. This method is loop-free, because the primary routes are hierarchical shortest path routes, while the secondary (alternate) routes essentially create a hierarchy within layer k. For example, Fig. 13 shows 4 nodes numbered 1301-1304 in layer k connected to 4 nodes numbered 1311 - 1314 in layer (k-l).

In Fig. 13, routing choices marked 1 are primary routes and routing choices marked 2 are secondary routes. It is clear that no inter-layer looping is possible since there is no downward routing - a node in layer (k-l) cannot route to a node in layer k. No intra-layer looping is possible, because node 1304 cannot alternate route to any of the other (lower numbered) nodes. This first method is simple to implement, but has the disadvantage that the highest numbered node (e.g., node 1304) in every layer is denied an alternate route. This disadvantage is overcome, albeit at the cost of added complexity, in the second method.

In the second method, we require two additional pieces of information These are:

(i) the ability to avoid single link loops of the form shown in Fig. 14, wherein node i must recognize that a packet was routed to it by node j, and must prevent the packet from going back to node j. This is necessary to avoid looping between nodes i and j when i and j route to each other on a second choice basis. As shown in Fig. 15, i.e., when the primary paths out of both nodes i and j are unavailable (due to congestion), packets must not be allowed to loop between i and j, but should be dropped.

(ii) Every node i must be assigned weights w(i,j) with respect to all other nodes j to which it is connected. Further, the weights must be chosen so that:

(a) they are symmetric, i.e., w(i,j) = w(j,i) for all i and j, and

(b) w(i,j) + w(j,k) is unique, in the sense that w(i,j) + w(i,k) = w(i,l) + w(l,m) => j=l and k=m.

This condition means that for any two nodes that have at least one two-hop path connecting them, there is a unique minimum weight two-hop path connecting the two nodes. One way of satisfying condition (b) is to choose weights so that the pairwise sum is unique, i.e., such that no two sums are the same. As will be described below, the weight information can be transmitted to each node together with the exclusionary tree routing information. The reason why the assignment of these weights is necessary will be also explained below.

Under the above conditions, the fact that our alternate routing technique is loop-free can be demonstrated as follows:

Let nodes 1,2,...m be in layer k and nodes 1', 2', 3',.....m ' in layer (k-l). The nodes in layer (k-l) may be repeated and are not necessarily unique (for notational convenience). Let us suppose that node i in layer k is connected to node i' in layer (k-l). There is no loss of generality in doing this because, even if layer (k-l) has a single node, it can be repeated m times. For example, Fig. 16, which shows links between nodes 1-3 of layer k and node 4 of layer k-l may be redrawn as Fig. 17, in witch node 1 is linked to node 1', node 2 is linked to node 2' and node 3 is linked to node 3', as long as nodes 1' ,2' and 3' are each "equal" to node 4.

The route $i \rightarrow i'$ is always the primary route from node i, for all packets to a particular destination D (with respect to which the network has been layered). With this notation in mind, our loop-free alternate routing technique may be expressed in the following manner:

Routing Rule: Let nodes i,j and ℓ belong to layer k and nodes j' and ℓ' belong to layer k-1. Then, node i alternate routes to node j if and only if

$$\min_{\ell \ \epsilon \ \text{layer} \ \kappa} \left\{ w(i,\ell)+w(\ell,\ell') \right\} = w(i,j)+w(j,j') \qquad (1)$$

Equation 1 is illustrated diagrammatically in Fig. 18, in which nodes i, j and j' are shown. In that figure, the link between nodes i and j is marked 2, indicating that this is the secondary path from node i to node j in layer k; the link between nodes j and j' is marked 1, indicating that this is the primary path from node j in layer k to node j' in layer k-1. In accordance with our technique, w(i,j)+w(j,j') is then the unique minimum weight 2-hop path to get from node i to any node in layer (k-l).

The loop-free property of our technique can be demonstrated by first considering the case when m=3, i.e., thsee nodes in layer k. Assume that nodes in layer k are fully connected. The corresponding network is shown

in Fig. 17. The connectivity between nodes in layer (k-1) is not important, and, hence, is not shown. The only possibility of looping occurs if each node in layer k alternate routes to a node in layer k that has not previously served as an alternate. For example, the situation illustrated in Fig. 19 is a loop, because node 1 alternate routes (marked "2") to node 2, node 2 alternate routes to node 3, and node 3 alternate routes back to node 1. Using our technique, such a loop cannot occur, because if node 1 alternate routes to node 2, and node 2 alternate routes to node 3, then node 3 must necessarily alternate route to node 2, so that a loop cannot occur. Now, the fact that node 1 alternate routes to node 2 implies that:

$$w(1,2) + w(2,2') < w(1,3) + w(3,3') \quad (2)$$

Next, the fact that node 2 alternate routes to node 3 implies that

$$w(2,3) + w(3,3') < w(2,1) + w(1,1') \quad (3)$$

Adding inequalities (2) and (3) and noting that w(i,j) = w(j,i), we get

$$w(3,2) + w(2,2') < w(3,1) + w(1,1') \quad (4)$$

which implies that node 3 alternate routes to node 2.

Thus, a three link loop cannot occur. However, a single-link loop may occur and hence the nodes must have the ability to recognize and prevent a single-link loop. Such a capability can be implemented simply in each node by preventing messages or packets from departing from the node on the same link that they arrived on. This is discussed in more detail below.

It should be noted that if the weights w(i,j) are chosen to be the actual distance d(i,j) between nodes i and j, then our invention leads to shortest distance alternate routing, which would be very important in a geographically dispersed network. However, while the symmetry property, viz., d(i,j) = d(j,i) is satisfied, the uniqueness property is not guaranteed. To ensure uniqueness, the internodal distances may have to be infinitesimally perturbed so that if

$$d(i,j) + d(j,j') = d(i,k) + d(k,k'), \quad (5)$$

then d(i,j) is changed to d(i,j) + ε, where ε is an arbitrary small number. However, it may be noted that since d(i,j) are real numbers, practically speaking the uniqueness condition is generally satisfied. The distance information can easily be provided to each node when the distributed shortest path route is determined, by providing V,H coordinates. If V_iH_i and V_jH_j are coordinates for two connected nodes i and j, the distance d(i,j) is given by

$$d(i,j) = \frac{\sqrt{(V_i - V_j)^2 + (H_i - H_j)^2}}{\sqrt{10}}$$

Other fully distributed techniques can be used to find a mapping between the nodes, i, j and j' used in alternate routing and the weights w(i,j),w(i,j') associated with the links between the nodes. For example, if each node, i, j, j' has a unique integer number, then w (i, j) can be arbitraly defined as (i^q+j^q) and w (i,j') can be likewise defined as $(j^q+j'^q)$, where q is a suitably chosen integer. Other mappings (i, j) → w (i, j) such that w (i,j) = w (j, i) and $w(i,j)+w(j,j')=w(i,\ell)+w(\ell,\ell')\Rightarrow j=\ell$ can also be found. However, weight assignments may also be centrally administered, and the appropriate weights periodically downloaded to each node when there is a topographical change in the network, without significantly degrading the performance of the network.

It is clear from the previous discussion that the topological information needed to construct loop-free alternate routing in accordance with our invention is the layering of the network with respect to every destination node. This information can be readily obtained from the exclusionary tree information which is already available in each node. Consider the layering with respect to destination node D shown in Fig. 20. Suppose that the node at which we are constructing the routing table is node S in layer k. Let node S be connected to nodes $S_1, \cdots S_k$ in layer k. Now, node S knows through its own primary routing table (constructed using exclusionary tree routing) that it is in layer k with respect to D. It also knows from the exclusionary trees received from $S_1, \cdots S_k$, that they are also in layer k with respect to D. There may be other nodes in layer k which S does not know about. But this does not matter as S is not connected to those nodes and hence could not alternate route to any of them. The key is that the exclusionary tree information is sufficient for a node to determine which of its neighbors are in the same layer as itself with respect to any given destination node. (This is unlike a centralized algorithm, in which all nodes have global knowledge of the network topology and, hence, every node knows all the other nodes in its layer. This is more information than needed, since a node only has to know the other nodes in the layer to which it is connected.)

The overall process by which each node determines its secondary routing table is shown in the flow chart of Fig. 21. Initially, in step 2101, network topology information, i.e., the identity of nodes that neighbor the current node is determined from the exclusionary tree routing information already available in the node. If another technique is used to generate the primary route, it is nevertheless assumed that this topology information is at hand. Likewise, in step 2102, the weights w_{ij} associated with the paths between the current node and its neighbors are computed from V, H coordinates if internodal distance is used as the weighing criteria, as described above. Otherwise, the appropriate weights are stored in the node.

Next, for a destinabon D, the network of nodes is organized into layers in step 2103, using the process

described in more detail in Figs. 22 and 23. As stated previously, each layer contains all nodes having the same distance, in hops, to the destination, using the shortest available path.

In step 2104 an alternate route to destination D is generated, using the process described in more detail in Figs. 24-25. This process is distributed, i.e., it is performed in each node independently.

After the alternate route for a given destination has been determined, a decision is made in step 2105 as to whether all destinations have been processed. If not, steps 2103 and 2104 are repeated for the other destinations. If routes for all destinations have been determined, the process stops in step 2106.

The layer generation or organization step 2103 of Fig. 21 is illustrated in more detail in Figs. 22 and 23. Initially, in step 2201, the identity of each of the neighbors of the current node i are stored in a memory or other suitable storage device. This information would be available at each node if the exclusionary tree routing process is used. In step 2202 the shortest distance (k) in hops, between i and the destination node D is computed. This information is also available as a result of the exclusionary tree routing process. It should be noted, however, that any other distributed shortest path algorithm can be used for primary route selection and any such algorithm would give us the shortest distance in hops. A similar procedure is then repeated for each neighbor j of i, in step 2203, to determine the distance m in hops between j and D. When the results of steps 2202 and 2203 are both available, a comparison between m and k is made in steps 2204 and 2214. If m and k are determined to be equal in step 2204, then it is concluded that j and i are in the same layer k (step 2205), and this information is stored (step 2206). If it is determined that m = k - 1 in step 2214, then it is concluded that j is in layer k-1 (step 2215) and this information is stored (step 2216). If the results of steps 2204 and 2214 indicate that m does not equal k or k-1, then it is concluded that j is in a more distant layer k+1 from D (step 2225). This information is therefore not needed, and is discarded in step 2226.

The layering process is further described in Fig. 23, which is a continuation of Fig. 22. After a particular neighbor j of the current node, i, has been examined to determine whether it is in the same layer k, a closer layer k-1, or a more distant layer k+1, a determination is made in step 2230 as to whether all neighbors j of node i have been examined. If not, the portion of the process beginning at step 2203 is repeated. After all neighbors of node i have been examined, a determination is made in step 2240 as to whether all destinations D have been examined. If not, the entire layer generation process, beginning at step 2202, is repeated for the next destination. When all destinations have been examined, the layering process is stopped in step 2250.

The alternate route generation process of step 2104 of Fig. 21 is described in more detail in Figs. 24 and 25. Initially for each destination D, all neighbors of node i that are in the same layer k as i (with respect to destination D) are stored in step 2401. This step thus uses the layering information previously obtained from the procedure described above in connection with Figs. 22 and 23. In a similar manner, all neighbors of node i in layer k-1 (with respect to destination D) are also stored, in step 2402. After information regarding the neighboring nodes has been stored, a determination is made in step 2403 of the weight w(i,j) associated with the link between nodes i and j within layer k. Similarly for each neighbor j' of node j in layer k-1, a determination is made in step 2404 of the weight w(j,j') associated with the link between node j in layer k and node j' in layer k-1. The sum of the weights w(i,j) and w(j,j') is next computed and stored in step 2405. At this point, a determination is made in step 2406 as to whether all neighbors j' of node j have been examined. If not, steps 2404 and 2405 are repeated for the next neighbor j'. After all neighbors j' have been examined, a determination is made in step 2407 (now referring to Fig. 25) as to whether all neighbors j of node i have been examined. If not, the computation process beginning with step 2403 is repeated. After all neighbors j of node i have been examined the stored combined weights are processed in step 2408 to select nodes j and j' such that w(i,j)+w(j,j') is a minimum. This minimum value determines the specific node j that is the alternate route for traffic from node i that is destined for node D.

After the alternate route for a specific destination D has been computed, a determination is made in step 2409 as to whether all destinations D have been processed. If not, the alterate route generation process beginning at step 2401 is repeated, so that a table of alternate routes, one for end destination, can be formed. When all destinations D have been processed, the process is stopped in step 2410.

In order to avoid the spread of congestion caused by alternate routing, another feature of our invention is the marking of a bit in the header of all packets that are routed on the alternate path. At all nodes in the alternate path, marked packets are given lower loss priority. This means that if the buffer occupancy at these nodes is below a preset threshold, then the marked packet is admitted, otherwise it is discarded. If the alternate path is also busy, then the alternate routed traffic is dropped and the spread of congestion is avoided. This process is illustrated in Fig. 26.

For each link outgoing from a node, a periodic measurement is made in step 2601 of the occupancy "x" of the buffer associated with that link. If x is determined to be less than a threshold value T_{alt} in step 2602, traffic on that link is uncongested, so that the uncongested routing table is selected in step 2603. If x is also less than a threshold value T_{accp}, both marked and unmarked packets are accepted for transmission over that link. How-

ever, if x is greater than or equal to T_{accp}, traffic on the link is relatively heavy (but still uncongested). In that circumstance, only unmarked packets are accepted in step 2607 for transmission over that link.

If it is determined in step 2602 that buffer occupancy x is equal to or greater than T_{alt}, the alternate route is selected in step 2604. However in this event, only a preselected fraction of packets are actually diverted from the primary route and routed on the alternate link. A test is next performed in step 2620 to determine if the outgoing link selected by alternate routing is connected to the same node as the node from which the packet was received. This test is performed in order to avoid single link loops, and is based on information relating to incoming and outgoing links that are readily available in the node. If the test result is positive so that a single link loop would be created, the packet is instead dropped or discarded in step 2621. Otherwise, a determination is made in step 2608 as to whether the alternate trunk was used to route the packet to the next node. If yes, the marking of that packet occurs in step 2609, so that the status of that packet as one having been alternate routed will be recognized in succeeding nodes. On the other hand, if alternate routing is not used, the packet is not marked (step 2610).

In accordance with another aspect of our invention, we have found it advantageous to use link buffer occupancy as a measure of link congestion, to determine when alternate routing should be applied, The activation and deactivation of alternate routing, as well as the decision to accept or reject an alternate routed cell, would then be based upon measurements of link buffer occupancy. Vinous specific buffer monitoring techniques can be used for this purpose, depending upon implementational convenience. For example, since link buffer occupancy fluctuates at great speed, it can be measured every millisecond. A running average of the 1000 most recent measurements can then be used to monitor congestion. When the average buffer occupancy exceeds a predetermined congestion threshold, some of the traffic is alternate routed, and these packets are marked by setting a loss priority bit in the header.

Fig. 27 illustrates, in simplified form, the functional architecture for a typical node 2701. As shown in that figure, node 2701 interconnects a series of incoming links 2710-2712 with a series of outgoing links 2720-2722. Links 2710-2712 and links 2720-2722 may in some implementation each be one or more high speed data trunks. Input buffers 2715-2717 receive packets applied on links 2710-2712, respectively, and apply the packets to a nodal processor 2730 to be described below. Likewise, output buffers 2725-2727 receive packets output from nodal processor 2730 that are destined for links 2720-2722, respectively. The occupancy or fullness of output buffers 2725-2727 are monitored in a congestion monitor 2740 which is part of nodal processor 2730, to determine when one or more links 2720-2722 is congested. The output of congestion monitor 2740 controls nodal processor 2730 such that a primary route to a destination is selected from table 2750 in the absence of congestion and an alternate route to a destination is selected from table 2760 in the presence of congestion. Nodal processor 2730 also includes a single link loop avoidance processor 2770, which is activated when congestion routing is used. The purpose of this processor is to assure that a packet originating at a neighboring node is not sent back to that node, so as to avoid forming a single link loop. This may be accomplished by keeping track of the input link on which a packet is received, and dropping the packet (i.e., not transmitting it) if the congested route specified by congested routing table 2760 is on a link back to the same node.

A more complete functional description of the arrangement of nodal processor 2730 is contained in Fig. 28. Nodal processor 2730 contains a central processing unit (CPU) 2810 and a memory 2850 having several portions. The network layering information that results from the process illustrated in Figs. 22 and 23 is stored for each destination node in portion 2802 of memory 2850, while network topology information is stored in another portion 2801 of the same memory 2850. Weights corresponding to different node pairs are also stored in the same portion of memory 2850.

Whenever there is a change in the network topology, the new network layering is calculated for each destination node and stored in portion 2802. CPU 2810 then uses the network layering information and the weight information to compute primary and alternate paths, which are stored in portions 2820 and 2830 of memory 2850. Persons skilled in the art will recognize that virious implementations for CPU 2810 and memory 2850 are readily available.

The benefits afforded by our alternate routing technique can be illustrated using a simple 3 node model of Fig. 29, which permits computation of end-to-end blocking with and without alternate routing for various offered loads. Based on this analysis, we have determined that alternate routing provides very significant improvements in end-to-end blocking.

In Fig. 29, node 2901 has two traffic streams, one destined for node 2902 and the other for node 2903. The traffic destined for node 2902 has a mean arrival rate of λ_{12} and the traffic destined for node 2903 has a mean arrival rate of λ_{13}. Node 2902 has a single traffic stream with mean arrival rate λ_{23} destined for node 2903. Let us suppose that, n_{12}, n_{13}, n_{23} are buffers in which cells from the traffic streams corresponding to λ_{12}, λ_{13} and λ_{23} queue up for service. All queues are first-in, first-out (FIFO.) All arrivals are assumed Poisson and all service times are exponential. It is assumed that there is no receive buffer overflow and, hence, we do not model

the receive buffers.

Using this model, the impact of alternate routing on the λ_{13} traffic is examined by subjecting a fraction of the λ_{13} traffic to alternate routing so that, if the occupancy in buffer n_{13} exceeds a certain specified threshold, called the rejection threshold, the alternate routable fraction of λ_{13} is offered to buffer n_{12}, for transmission through node 2. Buffer n_{12} accepts the alternate routed traffic only if its occupancy is below a specified threshold, called the acceptance threshold; if not, it gets rejected and is lost. Once the alternate routed traffic reaches node 2902, it is accepted by buffer n_{23} only if its occupancy is below the acceptance threshold. It is important to note that the node 2901 to 2902 and node 2902 to 2903 traffic streams are not subject to alternate routing. This is because we wish to study the impact of alternate routing on the end-to-end blocking of the node 2901 to 2903 traffic, as we increase λ_{13} while keeping λ_{12} and λ_{23} constant. The queueing model corresponding to the network in Fig. 29 is shown in Fig. 30.

In Fig. 30, λ_{13d} denotes the direct routed component of λ_{13}, and λ_{13a} denotes the alternate routable portion of λ_{13}. The overall arrival rate for the node 2901 to 2903 traffic is $\lambda_{13} = \lambda_{13d} + \lambda_{13a}$. Using a birth-death process model, we have derived exact expressions for the end-to-end blocking suffered by the three traffic classes. In our analysis, we assumed a buffer size of 100 for n_{12}, n_{13} and n_{23}, since it yields a cell blocking probability of roughly 10^{-6} at an offered load of 0.9. We chose the rejection threshold to be 70 and the acceptance threshold to be 50. This means that whenever the occupancy of buffer n_{13} exceeded 70, the cells from the λ_{13a} stream are alternate routed to buffer n_{12}. Buffer n_{12} accepts the alternate routed λ_{13a} cells only if its occupancy is below 50. Similarly, the alternate routed λ_{13a} cells are accepted by buffer n_{23} for transmission to node 2903 only if the occupancy at buffer n_{23} is below 50. All cells that are not accepted are lost. In this simple model, we have not accounted for message retransmission. We kept the offered load due to λ_{12} and λ_{13} constant at 0.8 and vied the offered load due to λ_{13} from 0.5 to 2.0. 25% of the 1-to-3 traffic was subject to alternate routing. The end-to-end blocking suffered by the node 2901 to 2903 traffic at these various loads, with and without alternate routing, is shown in Fig. 31. Curve 3101 gives the blocking probability without alternate routing and curve 3102 gives the blocking probability with alternate routing. From Fig. 31, it is clear that there is substantial improvement in end-to-end blocking, with alternate routing. Fig. 31 does not exhibit the sharp increase in blocking that normally occurs with other alternate routing techniques that do not mark packets to avoid the spread of congestion, as advantageously provided in our invention. Fig. 32 is a rescaled version of Fig. 31 showing the end-to-end blocking experienced by the node 2901 to 2903 traffic when the offered load ranges from 0.8 to 1.2. Again, curve 3201 represents blocking probability without alternate routing and curve 3202 represents blocking probability with alternate routing. Fig. 32 clearly shows the dramatic improvement in end-to-end blocking for the node 2901 to 2903 traffic over a range of offered load of practical interest. Because direct routed traffic is given priority (alternate routed traffic is accepted only if the buffer occupancy is below 50), the node 2901 to 2902 and node 2902 to 2903 traffic suffer no significant performance degradation, even when the offered load due to the node 2901 to 2903 traffic is 2.0. The end-to-end blocking for the node 2901 to 2902 and node 2902 to 2903 traffic remains virtually at zero.

In summary , the congestion control scheme in accordance with our invention has the following properties:

(a) guarantees loop-freedom;

(b) reacts to measurements and changes paths dynamically;

(c) needs local measurements only;

(d) does not spread congestion; and

(e) carries traffic on lightly loaded links.

Indeed, the invention allows a connectionless network to efficiently carry as much traffic as possible, since packet loss that ordinarily results from buffer overflow is reduced, and the retransmission problem is alleviated. No additional signaling messages need to be exchanged between network nodes.

Various modifications and adaptations of the present invention will be readily apparent to those of ordinary skill in the art. Accordingly, it is intended that the invention be limited only by the appended claims.

Claims

1. A method of routing information packets from a first node in a network of interconnected nodes to a destination node, comprising the steps of

   a) forming a first routing table containing the primary route to be taken by information packets at said first node destined for said destination node and a second routing table containing an alternate route to be taken by information packets when said primary route is congested;

   b) monitoring congestion in said network; and

   c) routing a portion of said information packets over said alternate route in the presence of congestion;

wherein said second routing table is formed by
d) determining other nodes in said network that are interconnected with said first node;
e) organizing each of said interconnected nodes including said first node into a series of layers in accordance with their distance, in hops, to said destination node;
f) assigning a weight to each possible path between said first node and each of said other interconnected nodes in the same layer,
g) assigning a weight to each possible path between each of said other interconnected nodes in said same layer and a connected node in a different layer, said different layer being closer to said destination node; and
h) selecting said alternate route by minimizing the pairwise sum of the weights obtained during said first and second assigning steps (f) and (g) above.

2. The method of claim 1 wherein said weight assigning steps include computing the distance between nodes using coordinate information representing the location of said nodes.

3. A method of controlling congestion in the flow of information bearing packets traveling over paths in a network of interconnected nodes, comprising the steps of
routing packets from each node to destination nodes via multihop primary routing paths;
monitoring congestion in said nodes in said network; and
routing packets from ones of said nodes to said destinations via alternate multihop routing paths in the event that congestion is encountered in said network;
wherein said alternate routing paths are determined by
grouping said interconnected nodes into a plurality of layers, each layer containing nodes that are the same distance, in hops from a particular destination;
assigning a weighting factor to each path between interconnected nodes in said layers;
assigning a weighting factor to each path between interconnected nodes in adjacent layers; and
selecting said alternate routing paths as a function of combinations of said weighting factors.

4. The invention defined in claim 3 wherein said primary path contains k hops and said alternate path contains at least k+1 hops.

5. The invention defined in claim 3 wherein said selecting step includes
forming the pairwise sum of weighting factors assigned during both of said assigning steps.

6. The invention defined in claim 3, wherein said assigning step includes:
forming said weighting factor as a function of the distance between nodes connected via said paths.

7. The invention defined in claim 3, wherein said alternate route is used only for a portion of the packets intended for a congested primary routing path.

8. The invention defined in claim 7, wherein said method further includes the steps of
marking any packet transmitted over an alternate routing path;
examining each packet at each node before it is routed, to determine if it has been marked; and
routing marked packets only if said node is uncongested.

9. A method of selecting loop free alternate multi-hop paths for information bearing packets traveling over a network of communication nodes, comprising the steps of
storing in each of said communication nodes information describing the connections between each node in said network and neighboring nodes;
storing in each of said communication nodes information for assigning weights assigned to paths between each connected pair of nodes;
grouping interconnected nodes into k layers, each layer containing nodes having the same distance, in hops, from a potential destination;
computing, for each node in layer k, the poise sum of the stored weights assigned to a) paths between said node and a first set of connected nodes in layer k; and b) paths between said first set of connected nodes in layer k and a second set of connected nodes in layer k-1, and
selecting as the alternate route from said node in layer k to said potential destination, the path having the smallest of said pairwise sums.

10. The invention defined in claim 9, wherein said first storing step includes forming an adaptive minimum spanning tree representation of said network.

11. The invention defined in claim 9, wherein said second storing step includes storing coordinate information representing the horizontal and vertical location of each of said nodes with respect to a reference system.

12. A method of reducing congestion in a connectionless network including a plurality of interconnected nodes, comprising the steps of

associating with each node in said network, a primary route to be taken by at least a portion of traffic from said node destined for each destination node;

associating with each node in said network, an alternate route to be taken by traffic from said node destined for each destination node in the event that said primary route is congested;

monitoring congestion in said network, and

routing traffic on said alternate route in the event that congestion is detected;

wherein said first association step includes forming a k-hop route using adaptive minimum spanning tree routing; and

wherein said second association step includes forming a route having at least k+1 hops, based upon connectivity information locally available in said each node.

13. In a network of interconnected nodes in which packets are transmitted over a primary route determined by selecting the shortest path, in hops, between originating node and the destination node, a method of providing an alternate route in the event said primary route is congested, said method comprising the steps of

grouping nodes between said originating node and said destination node into a plurality of groups, such that the nodes in the k^{th} group are equally distant, in hops, from said destination node;

assigning a weight, w(i,j) to each path between nodes i and j in group k and a weight w(j,j') to each path between node j in group k and node j' in group k-1,

selecting said alternate path such that w(i,j)+w(j,j') is minimized.

14. A method of determining an alternate route for traffic in a connectionless network of nodes when the primary route between said nodes is congested, comprising the steps of

for each destination, grouping said nodes as a function of the distance of said node from said destination;

assigning a first weighting factor to each path between a node in one of said groups and each connected node in the same group, and a second weighting factor to each path between each of said connected nodes in the same group and other connected nodes in another of said groups; and

selecting said alternate route as a function of said first and second routing factors.

15. Apparatus for controlling congestion in the flow of information bearing packets traveling over paths in a network of interconnected nodes, comprising

a) means for monitoring congestion in primary and secondary routing paths within said network; and

b) means for routing packets from each node to destination nodes via multihop primary routing paths in the absence of congestion and for routing packets from ones of said nodes to said destinations via alternate multihop routing paths in the event that congestion is encountered in said primary routing paths;

wherein said routing means includes

means for grouping said interconnected nodes into a plurality of layers, each layer containing nodes that are the same distance, in hops from a particular destination;

means for assigning a weighting factor to each path between interconnected nodes in said layers, and for assigning a weighting factor to each path between interconnected nodes in adjacent layers; and

means for selecting said alternate routing paths as a function of combinations of said weighting factors.

16. The invention defined in claim 15 wherein said primary path contains k hops and said alternate path contains at least k+1 hops.

17. The invention defined in claim 15 wherein said selecting means includes

means for forming pairwise sums of said weighting factors for

a) paths between nodes in the same layer, and
b) paths between nodes in adjacent layers.

18. The invention defined in claim 15, wherein said assigning means includes:
means for forming said weighting factor as a function of the distance between nodes connected via said paths.

19. The invention defined in claim 15, wherein said routing means is arranged so that said alternate route is used only for a portion of the packets intended for a congested primary routing path.

20. The invention defined in claim 19, wherein said apparatus further includes
means for marking any packet transmitted over an alternate routing path;
means for examining each packet at each node before it is routed, to determine if it has been marked; and
means for routing marked packets only if said node is uncongested.

21. Apparatus for selecting loop free alternate multi-hop paths for information bearing packets traveling over a network of communication nodes, comprising
means for storing in each of said communication nodes (a) information describing the connections between each node in said network and neighboring nodes, and (b) information for assigning weights to paths between each connected pair of nodes;
means for grouping interconnected nodes into k layers, each layer containing nodes having the same distance, in hops, from a potential destination;
means for computing, for each node in layer k, the pairwise sum of the stored weights assigned to a) paths between said node and a first set of connected nodes in layer k; and b) paths between said first set of connected nodes in layer k and a second set of connected nodes in layer k-1, and
means for selecting as the alternate route from said node in layer k to said potential destination, the path having the smallest of said pairwise sums.

22. Apparatus for reducing congestion in a connectionless network including a plurality of interconnected nodes, comprising
means for associating with each node in said network a) a primary route to be taken by at least a portion of traffic from said node destined for each destination node, and b) an alternate route to be taken by traffic from said node destined for each destination node in the event that said primary route is congested;
means for monitoring congestion in said network, and
means for routing traffic on said alternate route in the event that congestion is detected;
wherein said associating means includes (a) means for forming a k-hop route using adaptive minimum spanning tree routing, and (b) means for forming a route having at least k+1 hops based upon connectivity information locally available in said each node.

23. In a network of interconnected nodes in which packets are transmitted over a primary route determined by selecting the shortest path, in hops, between the originating node and the destination node, apparatus for providing an alternate route in the event said primary route is congested, said apparatus comprising
means for grouping nodes between said originating node and said destination node into a plurality of groups, such that the nodes in the k^{th} group are aqually distant, in hops, from said destination node;
means for assigning a weight, w(i,j) to each path between nodes i and j in group k and a weight w(j,j') to each path between node j in group k and node j' in group k-1, and
means for selecting said alternate path such that w(i,j)+w(j,j') is minimized.

24. Apparatus for determining an alternate route for traffic in a connectionless network of nodes when the primary route between said nodes is congested, comprising
for each destination, means for grouping said nodes as a function of the distance of said node from said destination;
means for assigning a first weighting factor to each path between a node in one of said groups and each connected node in the same group, and a second weighting factor to each path between each of said connected nodes in the same group and other connected nodes in another of said groups; and
means for selecting said alternate route as a function of said first and second weighting factors.

FIG. 1

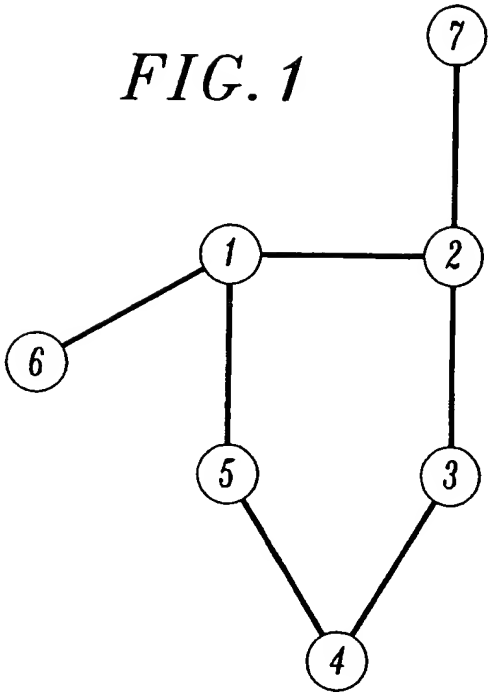


FIG. 2

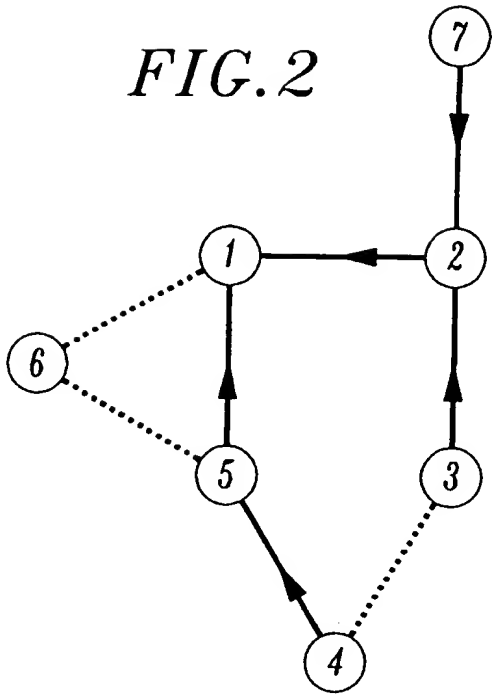


FIG. 3

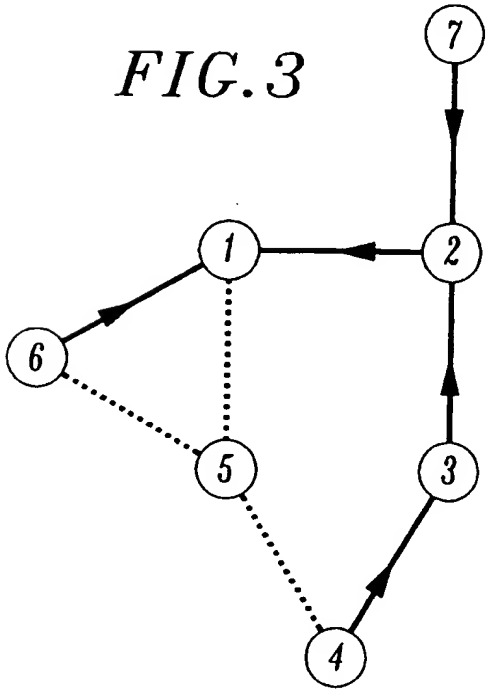


FIG. 4

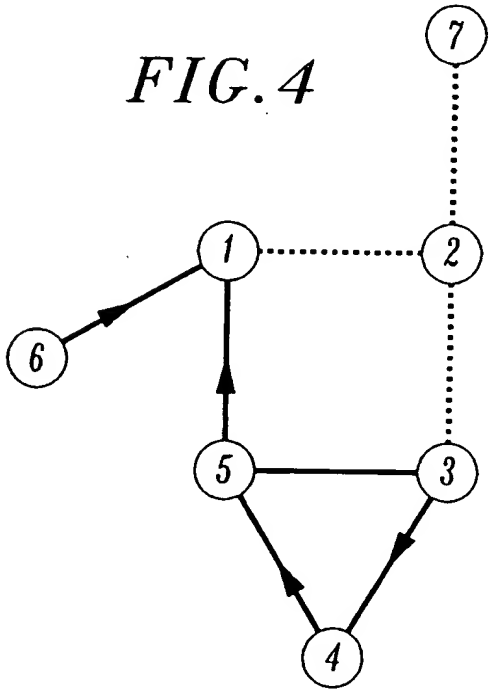


FIG.5

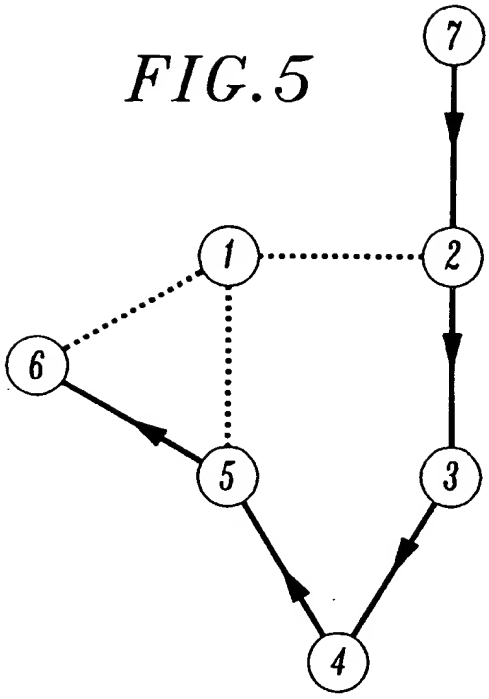


FIG.6

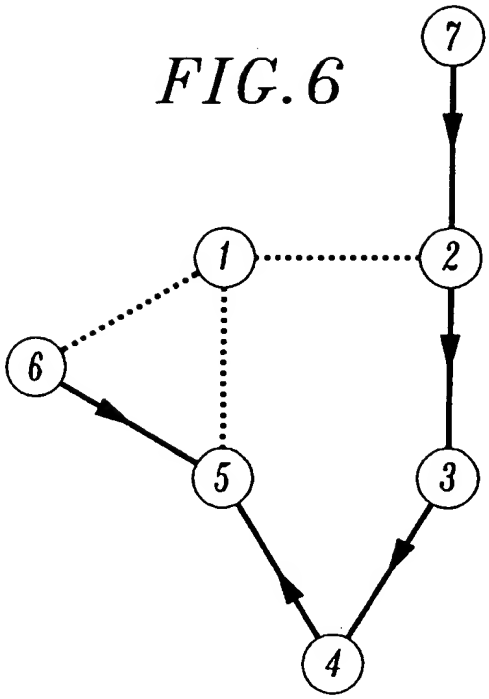


FIG.7

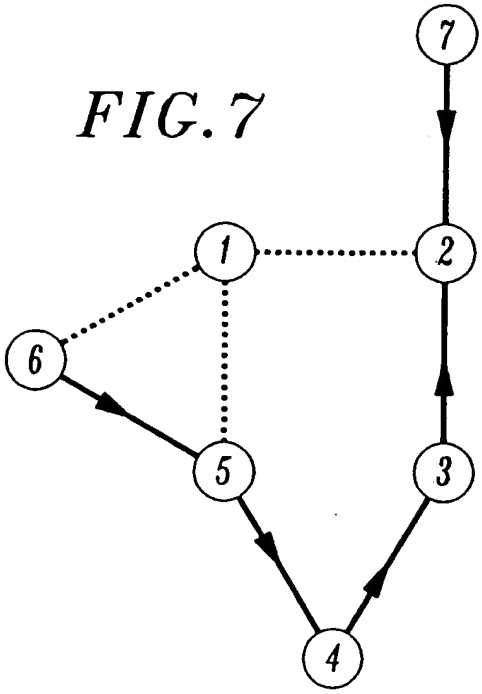


FIG. 8

FIG. 9

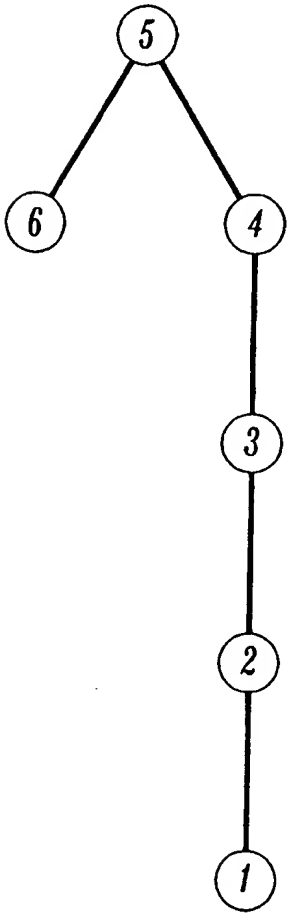


FIG. 10

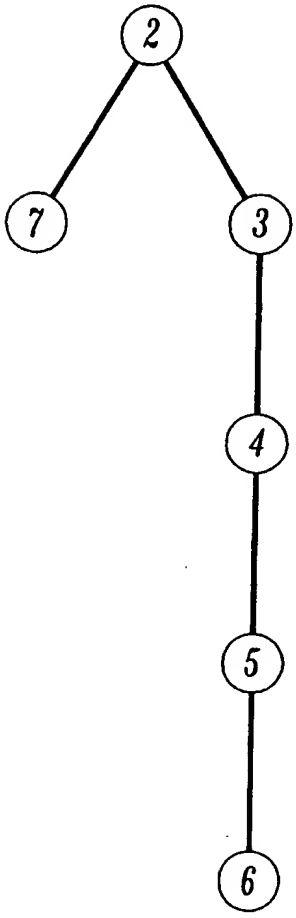
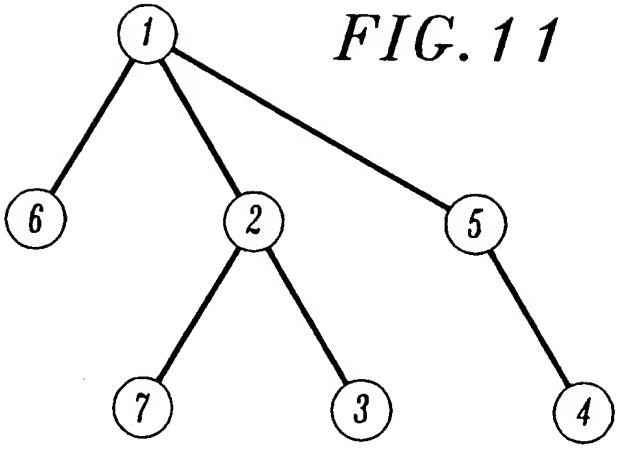


FIG. 11



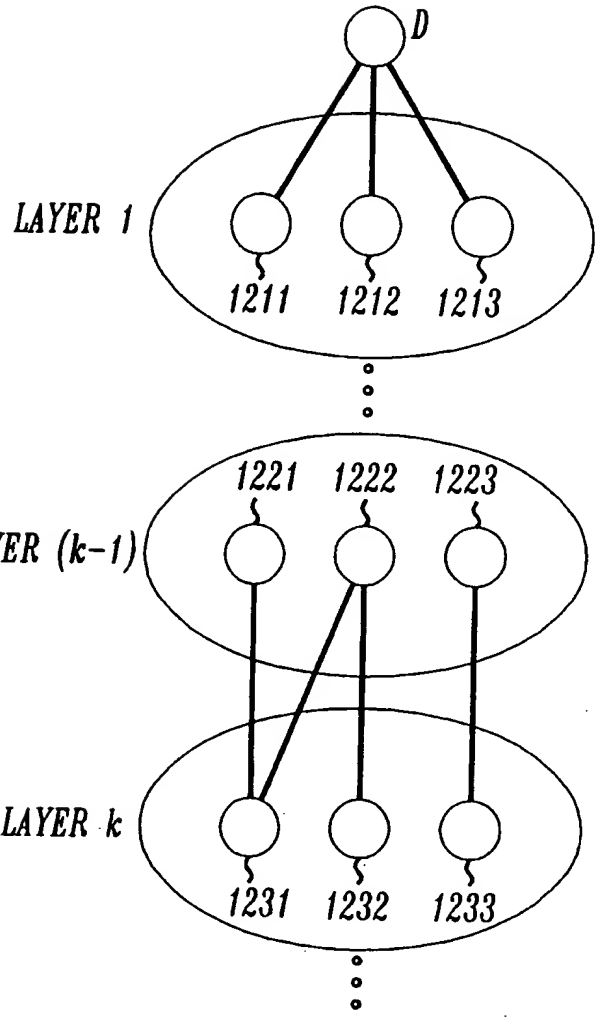

D
LAYER 1
1211
1212
1213
1221
1222
1223
LAYER (k-1)
LAYER k
1231
1232
1233

FIG. 12

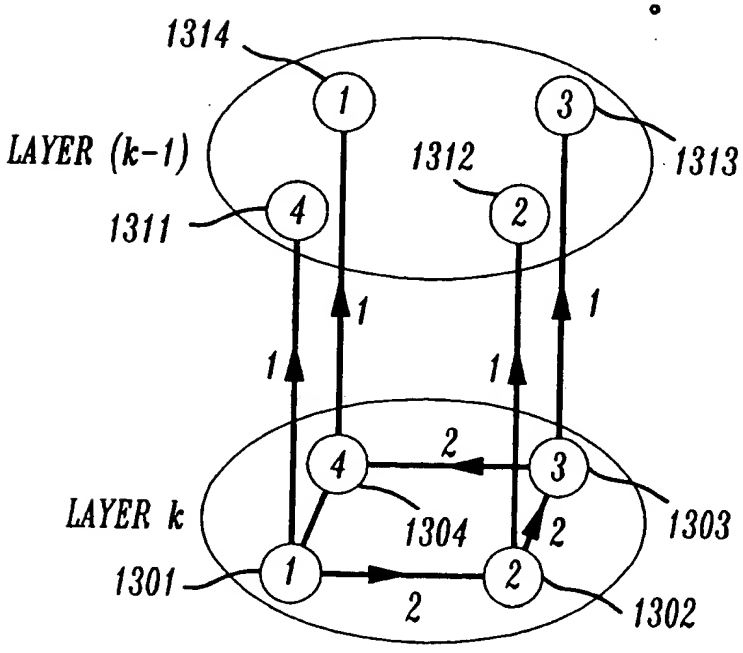

1314
1
3
LAYER (k-1)
1312
1313
1311
4
2
1
1
1
1
2
4
3
LAYER k
1304
2
1303
1301
1
2
2
1302

FIG. 13

FIG. 14

FIG. 15

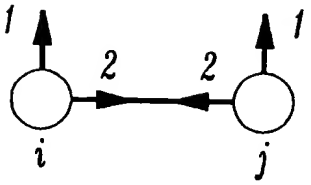


FIG. 16

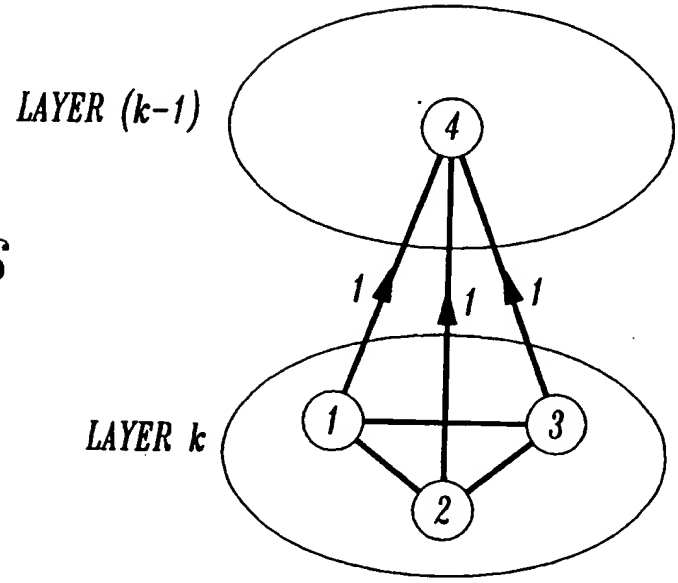


FIG. 17

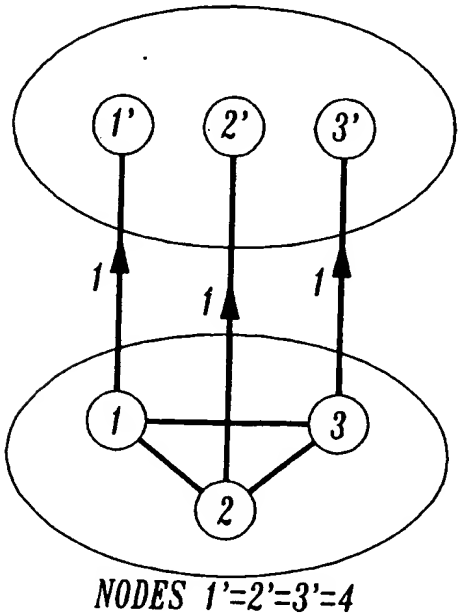


FIG. 18

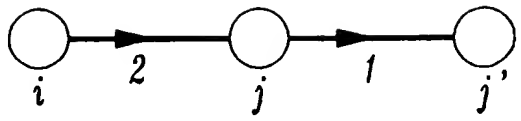


FIG. 19

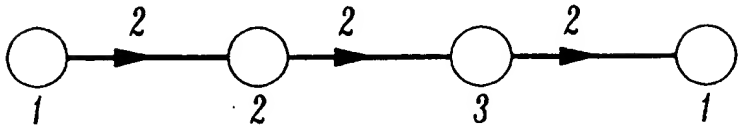


FIG. 20

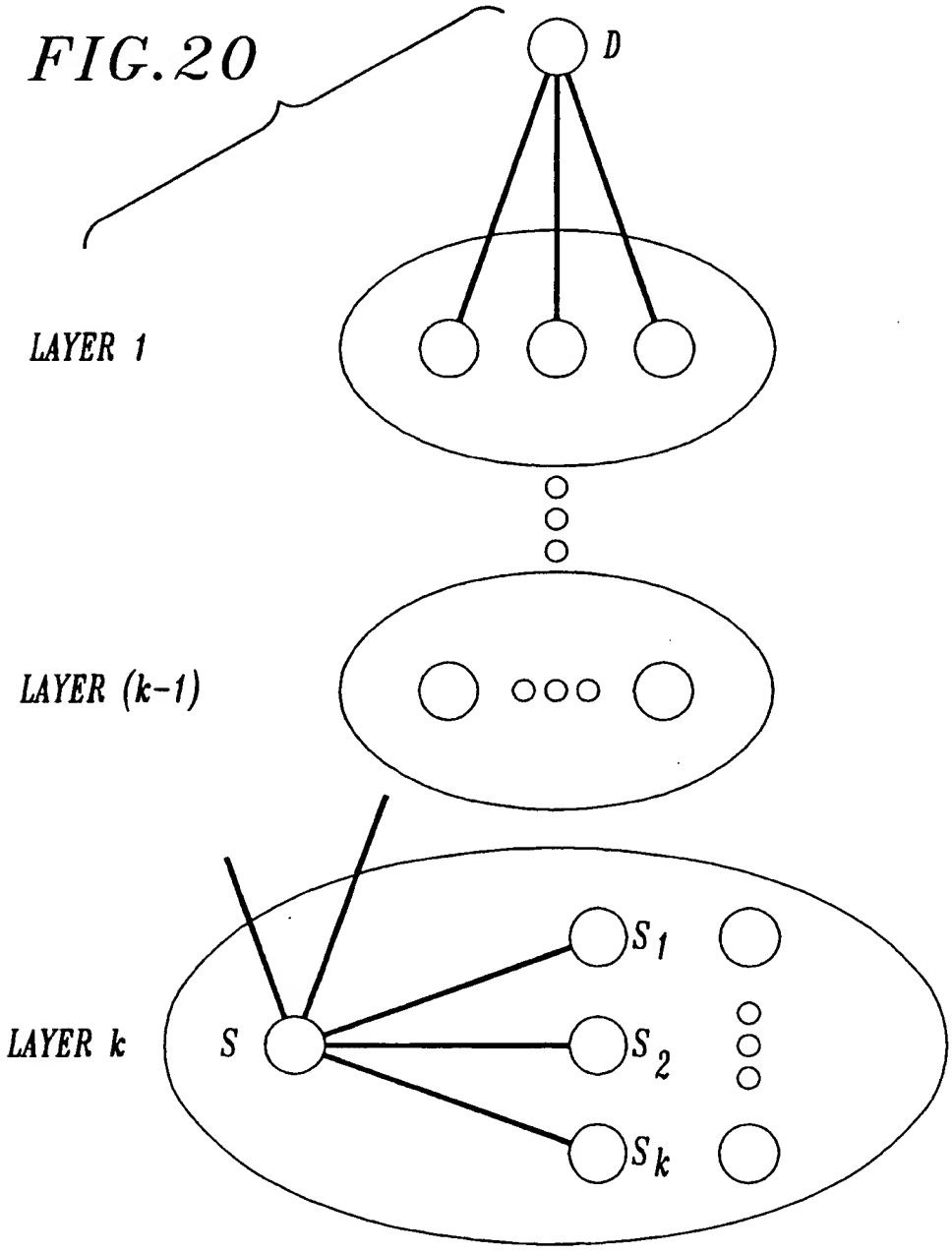
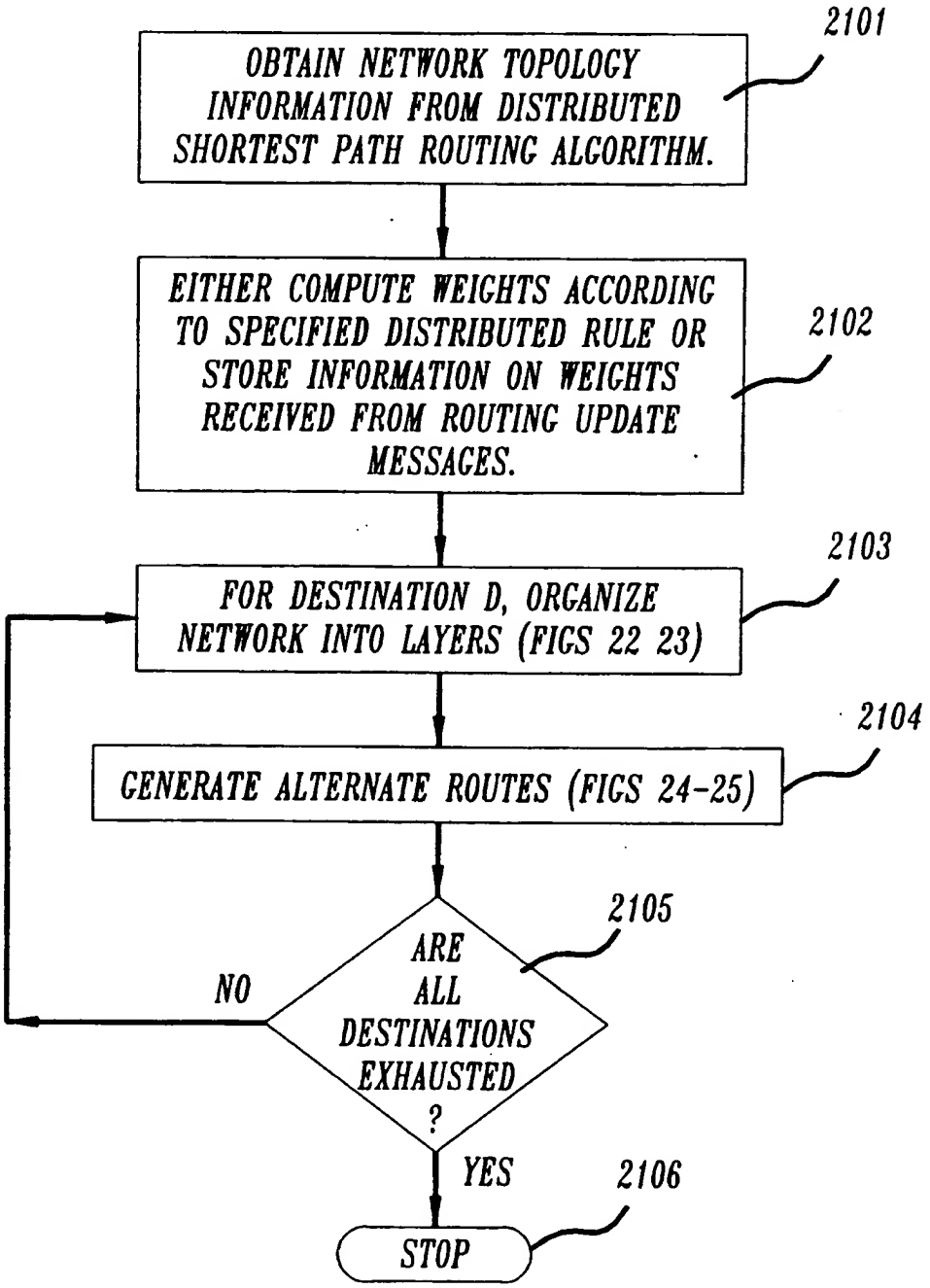


FIG. 21

FLOW CHART FOR ROUTING TABLE GENERATION



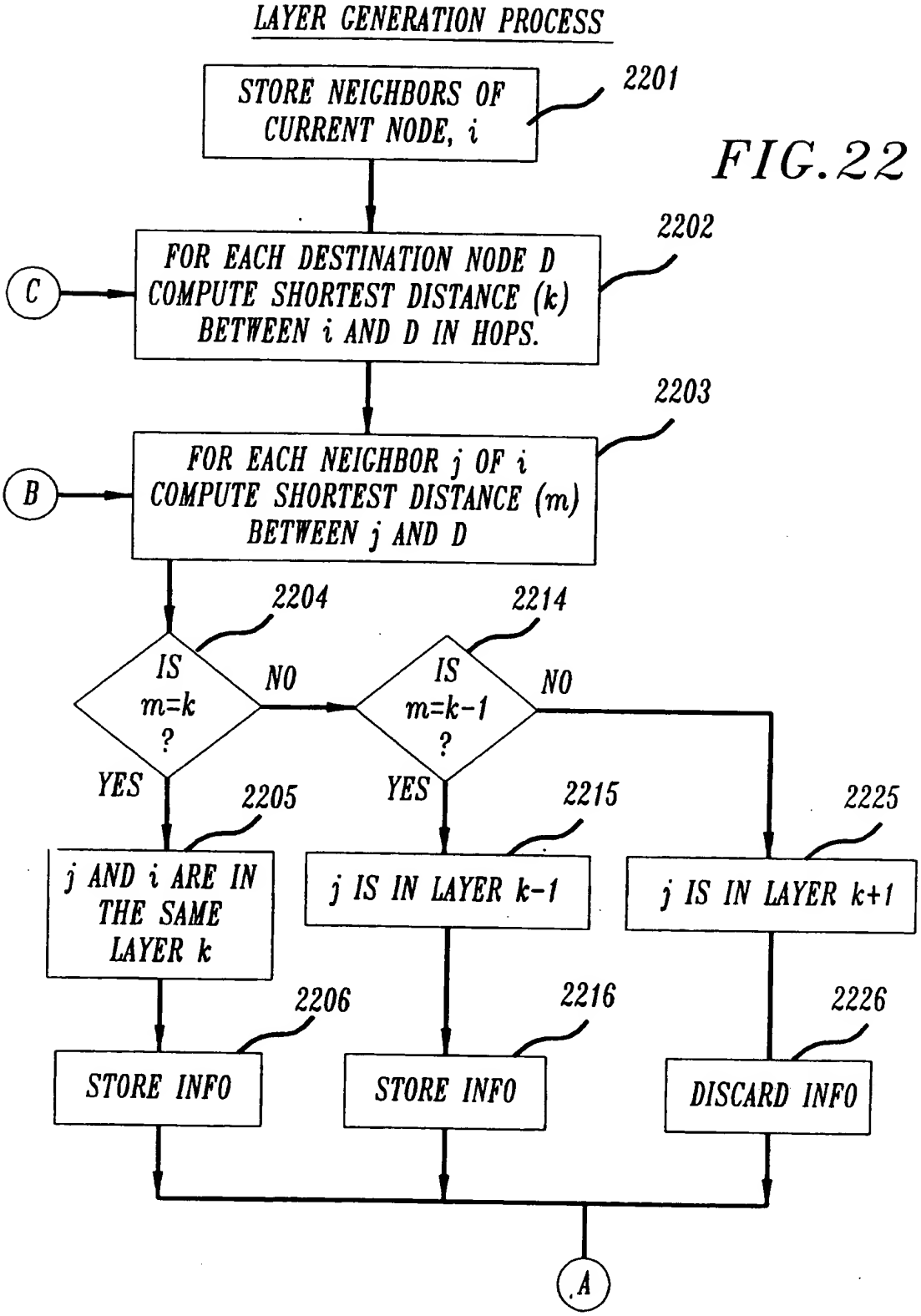

LAYER GENERATION PROCESS
STORE NEIGHBORS OF CURRENT NODE, i
2201
FOR EACH DESTINATION NODE D COMPUTE SHORTEST DISTANCE (k) BETWEEN i AND D IN HOPS.
2202
C
FOR EACH NEIGHBOR j OF i COMPUTE SHORTEST DISTANCE (m) BETWEEN j AND D
2203
B
IS m=k ?
2204
NO
IS m=k-1 ?
2214
NO
YES
2205
YES
2215
2225
j AND i ARE IN THE SAME LAYER k
j IS IN LAYER k-1
j IS IN LAYER k+1
2206
2216
2226
STORE INFO
STORE INFO
DISCARD INFO
A

FIG. 22

FIG.23

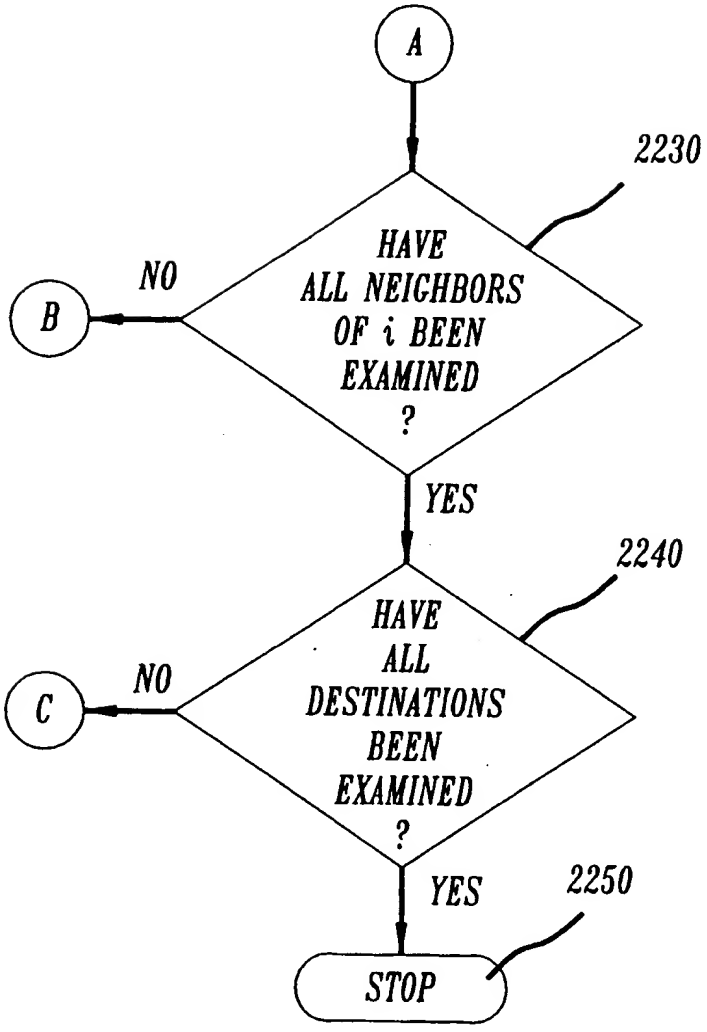

A
2230
HAVE ALL NEIGHBORS OF i BEEN EXAMINED ?
NO
B
YES
2240
HAVE ALL DESTINATIONS BEEN EXAMINED ?
NO
C
YES
2250
STOP

FIG. 24

ALTERNATE ROUTE GENERATION

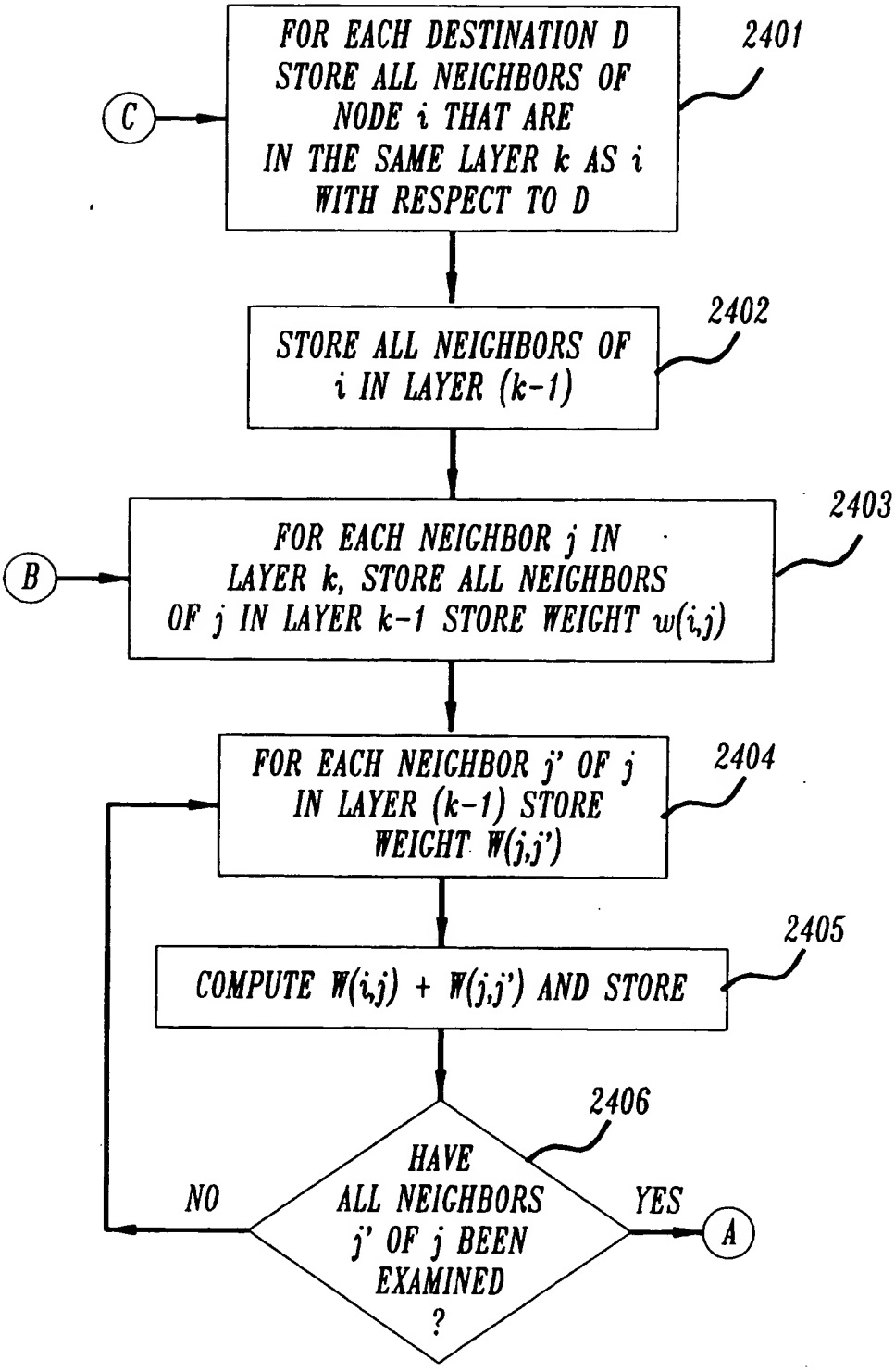
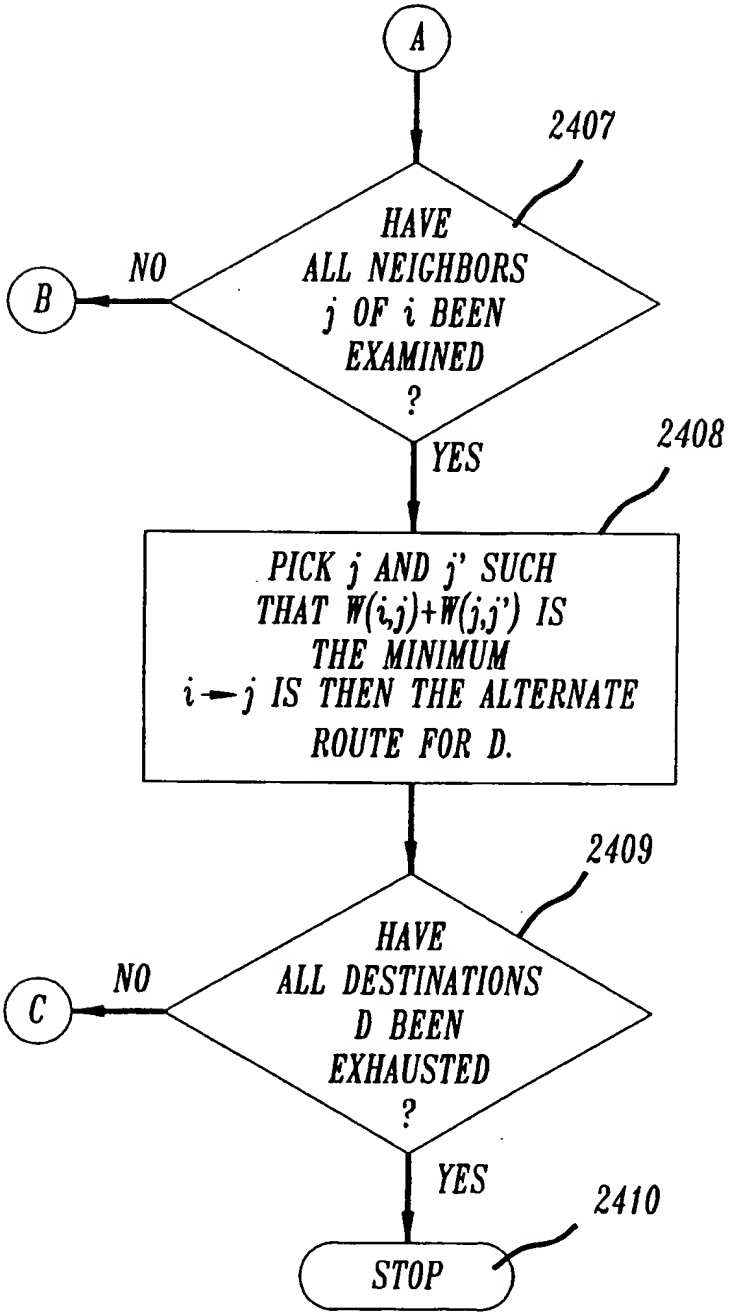


FIG. 25


A
2407
HAVE ALL NEIGHBORS j OF i BEEN EXAMINED ?
NO
B
YES
2408
PICK j AND j' SUCH THAT W(i,j)+W(j,j') IS THE MINIMUM i→j IS THEN THE ALTERNATE ROUTE FOR D.
2409
HAVE ALL DESTINATIONS D BEEN EXHAUSTED ?
NO
C
YES
2410
STOP

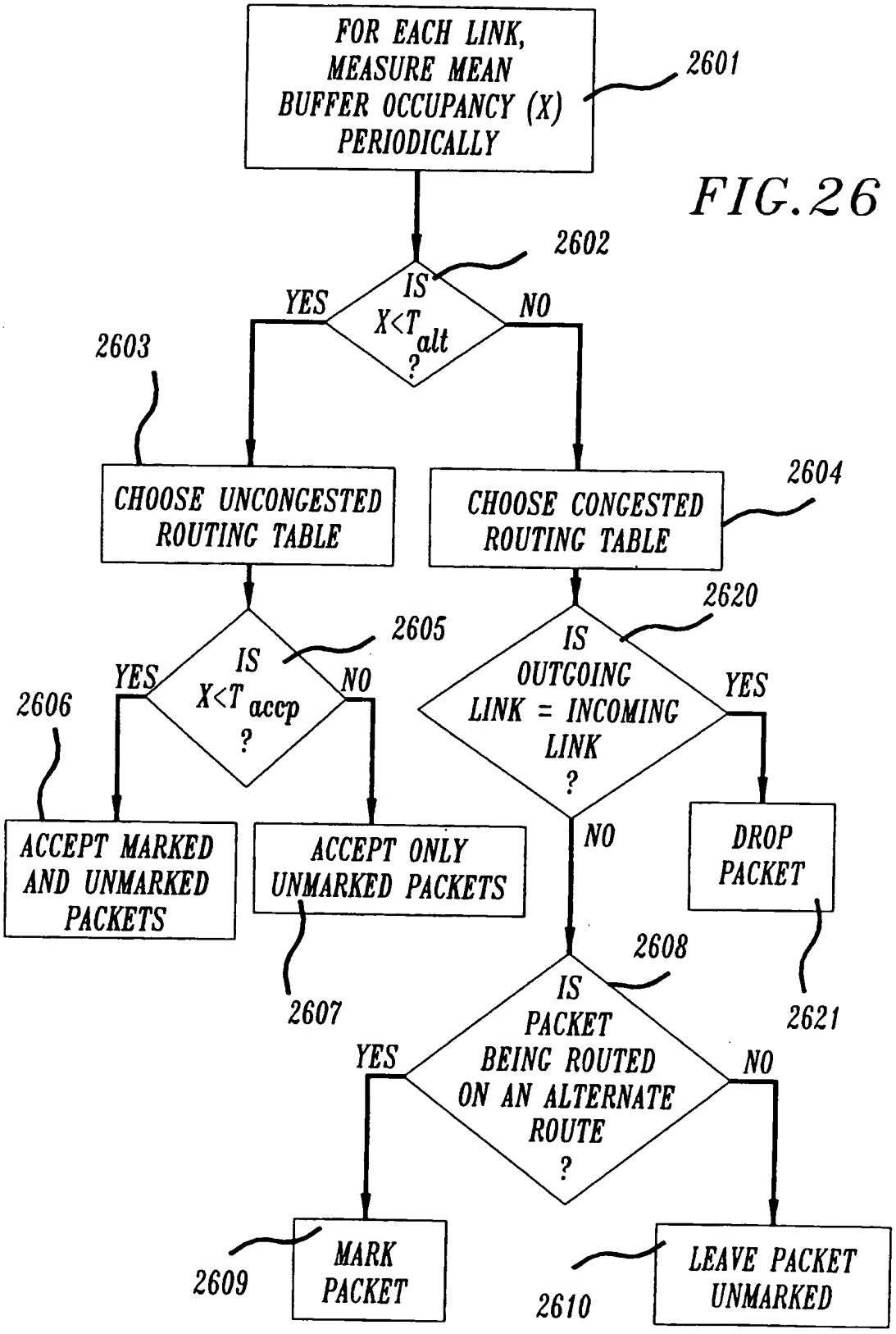

FOR EACH LINK, MEASURE MEAN BUFFER OCCUPANCY (X) PERIODICALLY
2601
2602
IS X<T alt ?
YES
NO
2603
CHOOSE UNCONGESTED ROUTING TABLE
CHOOSE CONGESTED ROUTING TABLE
2604
2605
IS X<T accp ?
YES
NO
2606
ACCEPT MARKED AND UNMARKED PACKETS
ACCEPT ONLY UNMARKED PACKETS
2607
2620
IS OUTGOING LINK = INCOMING LINK ?
YES
NO
DROP PACKET
2621
2608
IS PACKET BEING ROUTED ON AN ALTERNATE ROUTE ?
YES
NO
2609
MARK PACKET
2610
LEAVE PACKET UNMARKED

FIG. 26

FIG.27

FUNCTIONAL NODE ARCHITECTURE

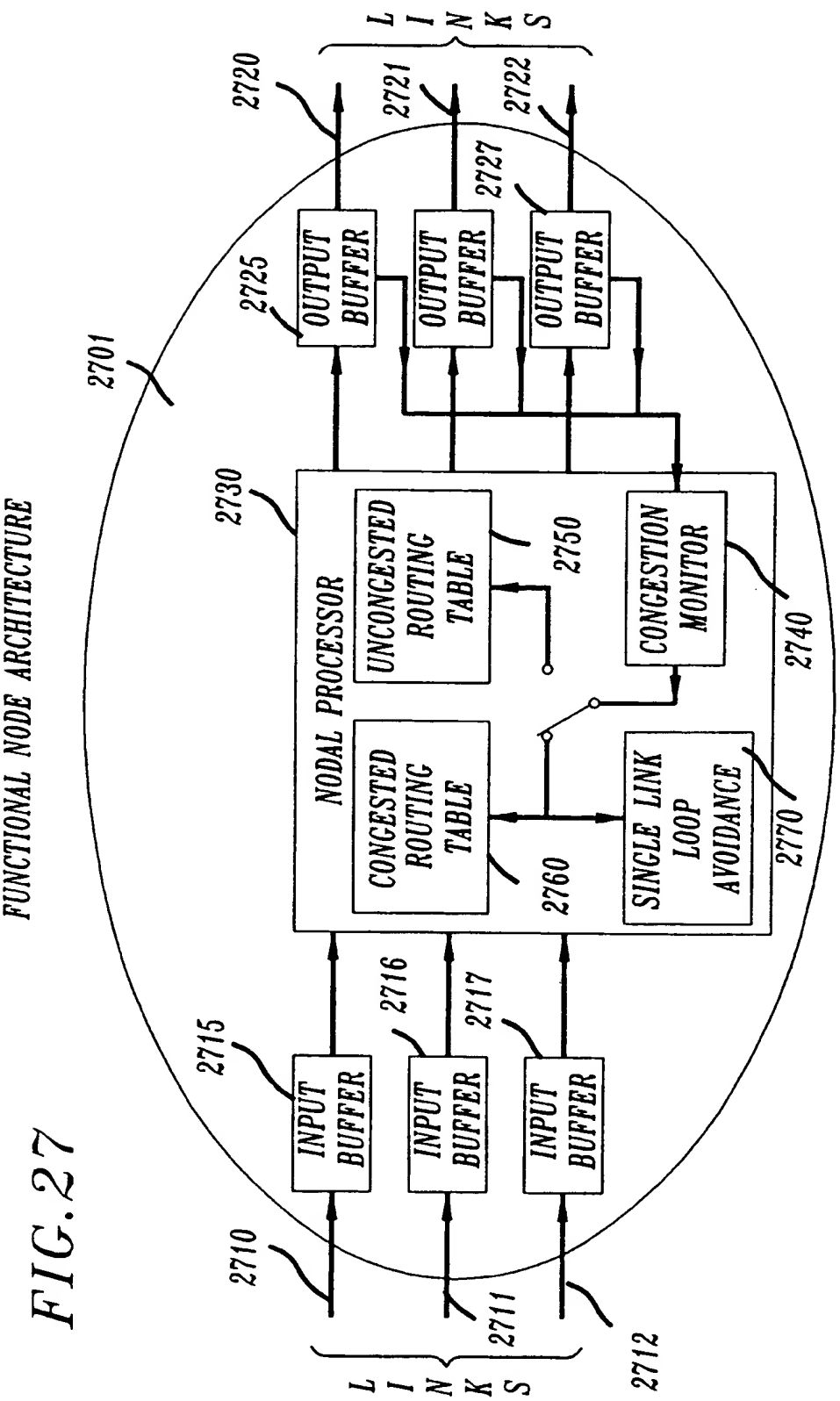


FIG.28

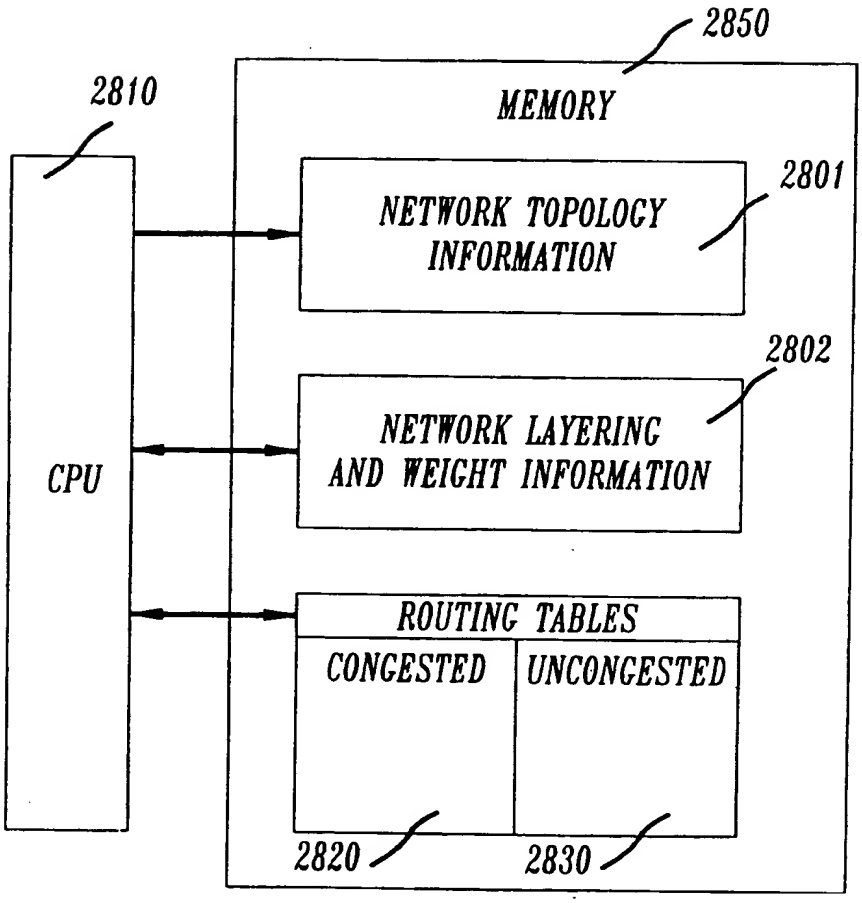

2850
MEMORY
2810
CPU
NETWORK TOPOLOGY INFORMATION
2801
NETWORK LAYERING AND WEIGHT INFORMATION
2802
ROUTING TABLES
CONGESTED
UNCONGESTED
2820
2830

FIG.29

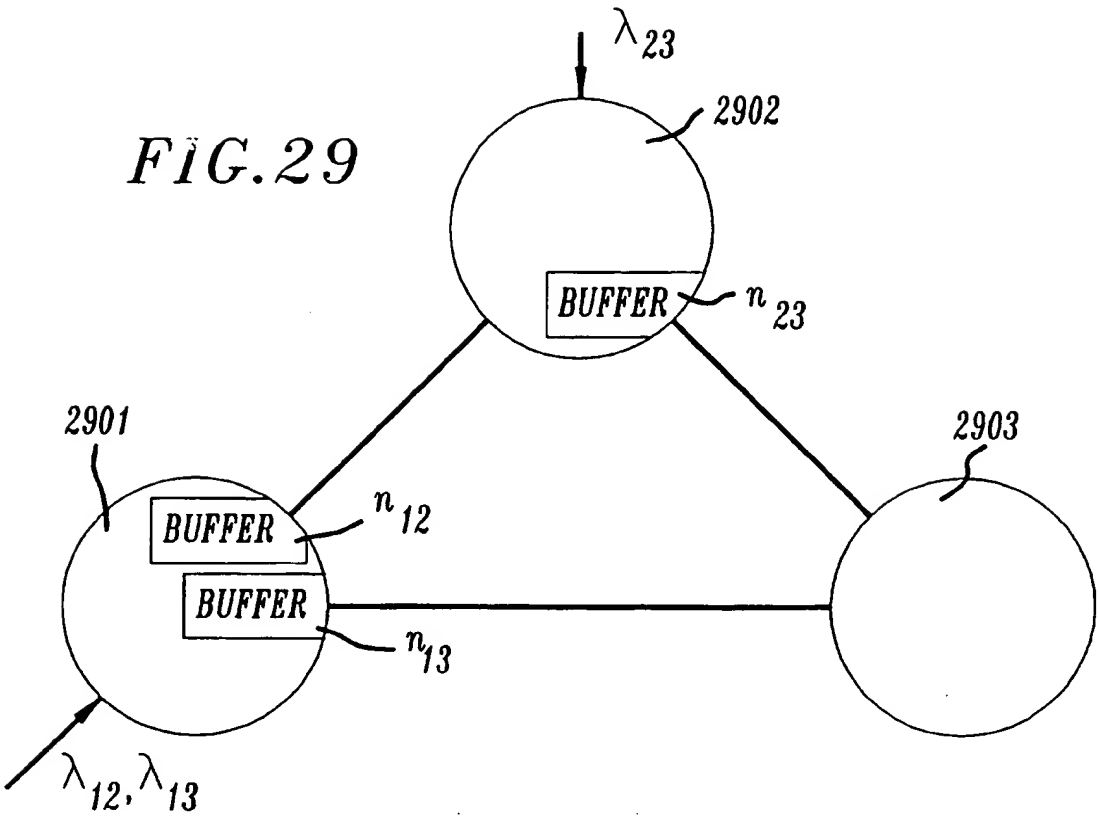

λ23
2902
BUFFER
n23
2901
2903
BUFFER
n12
BUFFER
n13
λ12, λ13

FIG.30

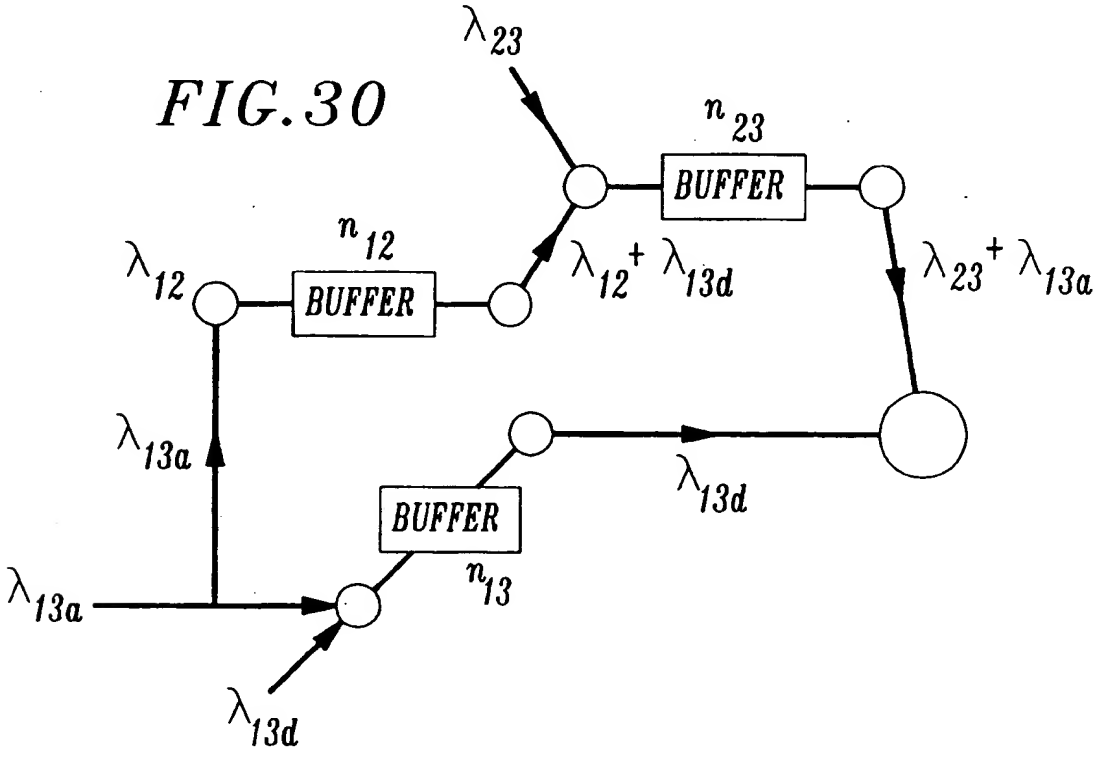

λ23
n23
BUFFER
n12
λ12
BUFFER
λ12+ λ13d
λ23+ λ13a
λ13a
λ13d
BUFFER
n13
λ13a
λ13d

FIG.31

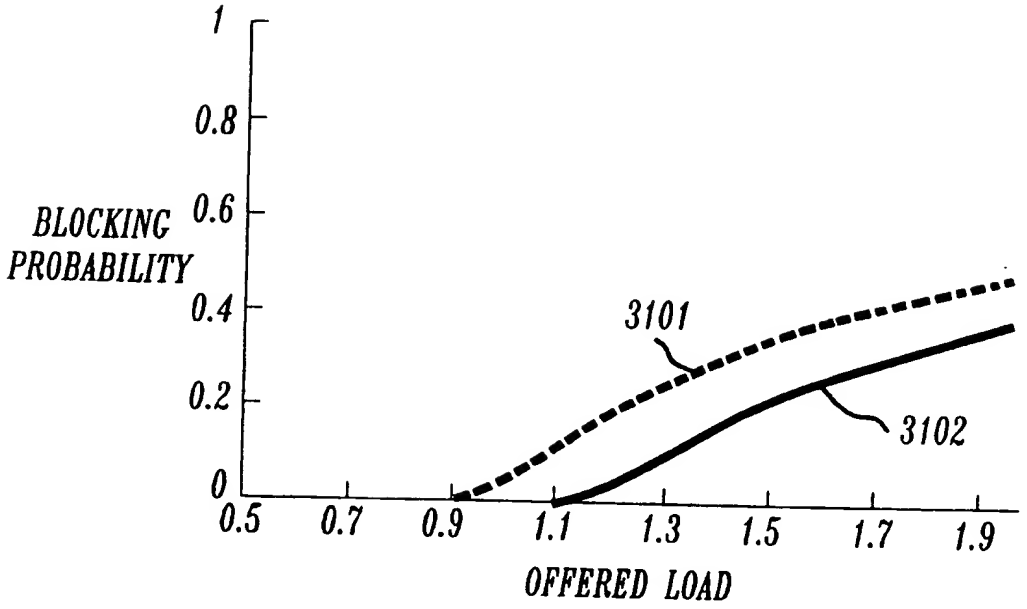
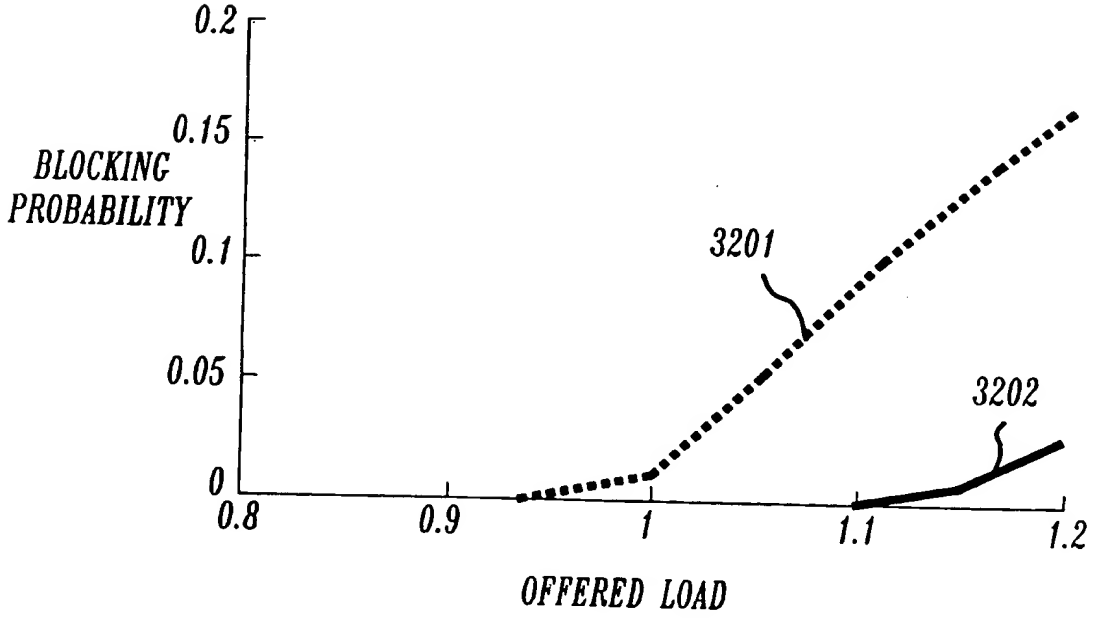

1
0.8
0.6
0.4
0.2
0
BLOCKING PROBABILITY
3101
3102
0.5
0.7
0.9
1.1
1.3
1.5
1.7
1.9
OFFERED LOAD

FIG.32


0.2
0.15
0.1
0.05
0
BLOCKING PROBABILITY
3201
3202
0.8
0.9
1
1.1
1.2
OFFERED LOAD

European Patent Office

EUROPEAN SEARCH REPORT

Application Number

EP 91 30 5728

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int. Cl.5)
A	GLOBECOM'82 December 1982, NEW YORK US pages 559 - 564; J.KAO ET AL.: 'Comparative evaluation of path routing strategies for mixed voice and data communication network ' * page 560, left column, line 29 - right column, line 6 * ---	1,3,4,7, 9,12-17, 19,21-24	H04L12/56
A	EP-A-348 328 (I.B.M.) * abstract; figure 1 * ---	1,3,9, 12-14, 15,21-24	
A	US-A-4 811 337 (J.H. HART) * abstract; figure 1 * ---	1,3,9, 10, 12-14, 15,21-24	
A	INFOCOM'90 June 1990, NEW YORK US pages 481 - 486; G.E. MYERS ET AL.: 'Routing in TAC-a triangularly arranged network ' * page 483, left column, line 5 - line 21 * ---	2,6,11, 18	TECHNICAL FIELDS SEARCHED (Int. Cl.5) H04L
A	EP-A-275 678 (A.T. &T.) * abstract * -----	8,20	

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	17 SEPTEMBER 1991	PEREZ PEREZ J.C.

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P0401)

Requested Patent: EP0859492A2

Title: FAIR QUEUING SYSTEM WITH ADAPTIVE BANDWIDTH REDISTRIBUTION ;

Abstracted Patent: EP0859492 ;

Publication Date: 1998-08-19 ;

Inventor(s):

DUFFIELD NICHOLAS G (US); STILIADIS DIMITRIOS (US); LAKSHMAN TIRUNELLAI VISWANATHA (US) ;

Applicant(s): LUCENT TECHNOLOGIES INC (US) ;

Application Number: EP19980300754 19980203 ;

Priority Number(s): US19970037844P 19970207; US19970972424 19971118 ;

IPC Classification: H04L12/56 ; H04L29/06 ;

Equivalents: JP10313324

ABSTRACT:

Apparatus for routing packets in a communication network comprises a plurality (100) of per-connection queues (20a--20i), each queue established for receiving packets from a respective source and temporarily storing received packets before routing to a particular destination; a weighted fair-queuing scheduler (40) for servicing packets from each of the plurality of per-connection queues at guaranteed pre-allocated rates; a sensing device for sensing a presence or absence of packets in queues, the absence of packets in queues indicating availability of excess bandwidth; and, a state dependent scheduler (200) for redistributing excess bandwidth upon sensing of queues absent packets, the state dependent scheduler (200) servicing those queues in accordance with a state variable corresponding to a performance property of the queues, wherein delay and isolation properties for routing packets of respective queues in weighted fair-queuing is preserved.

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 859 492 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
19.08.1998 Bulletin 1998/34

(51) Int. Cl.[6]: **H04L 12/56**, H04L 29/06

(21) Application number: **98300754.3**

(22) Date of filing: **03.02.1998**

(84) Designated Contracting States:
AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **07.02.1997 US 37844 P**
18.11.1997 US

(71) Applicant:
LUCENT TECHNOLOGIES INC.
Murray Hill, New Jersey 07974-0636 (US)

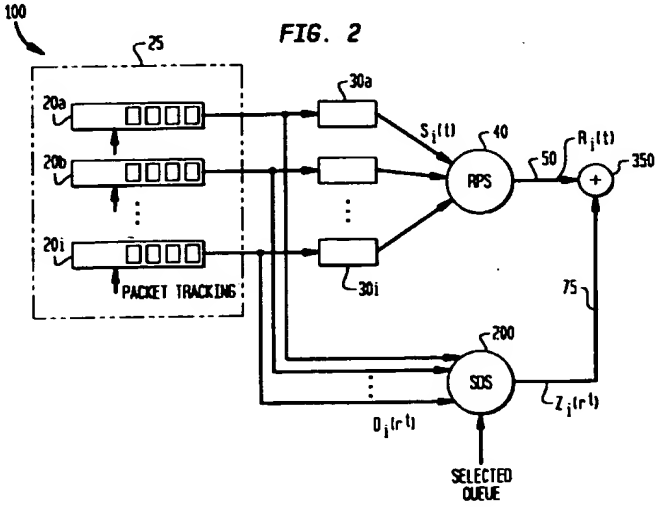
(72) Inventors:
- **Duffield, Nicholas G.**
  Hoboken, New Jersey 07030 (US)
- **Stiliadis, Dimitrios**
  Middletown, Monmouth, NJ 07748 (US)
- **Lakshman, Tirunellai Viswanathan**
  Eatontown, Monmouth, New Jersey 07724 (US)

(74) Representative:
Buckley, Christopher Simon Thirsk et al
Lucent Technologies (UK) Ltd,
5 Mornington Road
Woodford Green, Essex IG8 0TU (GB)

(54) Fair queuing system with adaptive bandwidth redistribution

(57) Apparatus for routing packets in a communication network comprises a plurality (100) of per-connection queues (20a--20i), each queue established for receiving packets from a respective source and temporarily storing received packets before routing to a particular destination; a weighted fair-queuing scheduler (40) for servicing packets from each of the plurality of per-connection queues at guaranteed pre-allocated rates; a sensing device for sensing a presence or absence of packets in queues, the absence of packets in queues indicating availability of excess bandwidth; and, a state dependent scheduler (200) for redistributing excess bandwidth upon sensing of queues absent packets, the state dependent scheduler (200) servicing those queues in accordance with a state variable corresponding to a performance property of the queues, wherein delay and isolation properties for routing packets of respective queues in weighted fair-queuing is preserved.

FIG. 2



EP 0 859 492 A2

Description

Field of the Invention

The instant invention relates generally to packet-based communication systems and particularly to fair-queuing systems implemented in routers and switches in a packet-based communication system.

Background of the Invention

Much research has been devoted to development of queuing systems for packet-based communication networks that emulates as closely as possible, an ideal "fluid flow" model, i.e., where data packets communicated from multiple sources are considered to be infinitely divisible and multiple sources may transmit their data simultaneously, e.g., on a single physical communication link. Infinite divisibility is not feasible in practice. In packet networks, typically once a packet is transmitted over the link, the whole packet must be sent, i.e., it cannot be interrupted to transfer another packet in between. As there exists a desire to provide Quality of Service "QoS" guarantees in a packet network, there is required the implementation of traffic scheduling methods in the data packet switches or routers. The function of a scheduling method is to select, for each outgoing link of a switch, the packet to be transmitted in the next cycle from the available packets belonging to the communication sessions sharing the output link. This selection must be performed such that the QoS guaranteed for the individual traffic sessions, e.g., upper bounds on maximum delay, are satisfied. Implementation of the method may be hardware or software, but because of speed considerations, scheduling is usually implemented in hardware in ATM switches and high-speed routers.

Many different scheduling methods have been proposed to approximate the theoretical scheduling discipline known as Generalized Processor Sharing (GPS) system, which is a discipline defined with respect to the "fluid" model. Such a GPS would allow for tight control of the bandwidth allocated to each session communicating on a link. However, as packets transmitted by a session cannot be divided further, the data from multiple sources must be interleaved only at packet boundaries. Thus, the GPS discipline cannot be implemented in practice in a packet-switched network.

Servicing of separate queues by simple FIFO, Round Robin, and fair queuing techniques, and the like, are well-known. However, "Weighted" fair-queuing ("WFQ") schemes have been developed that closely approximate the fluid system. Particularly, A. Demers, S. Keshav, S. Shenker, in the reference "Analysis and Simulation of a Fair Queuing Algorithm" *Internetworking: Research and Experience*, pp, 3-26, vol. 1, 1990 describe a fair queuing scheme that emulates GPS by essentially simulating a fluid flow GPS system for reference and basing packet scheduling decisions on the order of departures in the GPS system. In weighted fair queuing, each traffic session *i* sharing the output link controlled by the scheduling method is assigned a value ϕ_i corresponding to the reserved bandwidth of the session. The values ϕ_i are computed such that the reserved bandwidth of session *i* on the link is given by:

$$\frac{\Phi_i}{\sum_{j=1}^{\upsilon} \Phi_j},$$

where the denominator computes the sum of the ϕ_i, values for all υ sessions sharing the link.

Particularly, as shown in Fig. 1, a WFQ system 100 is provided with a plurality of per-connection queues 20a,...,20i, with each queue storing packets in a different portion of a shared memory 25 for temporarily storing packets of information, e.g., input traffic from a source device such as a data terminal. It is understood that there can be provided different types of queues for accommodating different types of traffic, e.g., audio, video, data, etc. Additionally provided is a shaper 30a,...,30i that forward packets from the queues to the Weighted Fair Queueing Server with a rate exactly equal to the allocated. The Weighted Fair Queueing scheduler assumes that a weight is associated with each queue 20a,..,20i; respectively, so that the service offered by the scheduler to each one of these queues while they have packets waiting is always in proportion to the weights. For example, let us assume that the capacity (bandwidth) of the link C=10 packets/sec. Let us also assume that the scheduler is serving three queues; Q1 being accorded a weight WQ1=20%, queque Q2 being accorded a weight WQ2=30%, and queue Q3 being accorded a weight WQ3=50%. Then, if all queues have packets waiting, then Q1 and Q2 will receive a guaranteed bandwidth of 2 and 3 packets/second respectively, and Q3 will receive a guaranteed bandwidth of 5 packets/second. However, if, for example, Q3 does not have any packets waiting, then the excess bandwidth is equal to 5 packets/second. In a WFQ system, this excess bandwidth is redistributed in proportion to the associated weights of the queues that have packets waiting. In the above example, when queue Q3 does not have packets waiting, the excess bandwidth will be distributed proportionally to queues Q1 and Q2 so, that they now receive instaneous bandwidth of 4 and 6 packets per second respectively. Each packet leaving its respective shaper 30 is forwarded directly to a Rate Proportional Server 40 ("RPS"), which may be any weighted fair queuing variation, that forwards the packets to output link 50.

In such a WFQ scheme, beneficial properties exist such as end-to-end delay guarantees, e.g., each packet is guaranteed a certain rate for each packet flow in the

stream, and, the provision of isolation between streams, e.g., a misbehaving source will not effect the flow of other streams. Additionally, an added benefit is that when there is underutilization of capacity, e.g., when flow is particularly bursty and there may be idle time, the WFQ system facilitates the redistribution of the unused bandwidth so as to preserve work-conservation property. Presently, the redistribution property of unused bandwidth capacity among the queues is done in a manner inherited from the fluid-flow model, e.g., in accordance with the weight associated with the particular queue. Thus, when the packet queues are idle, "excess" bandwidth is re-distributed to backlogged connections in proportion to their weights which are based on long-term requirements.

The drawback of GPS that all fair queuing systems inherit in their close emulation of GPS is that GPS severely restricts state-dependent bandwidth sharing. The only state-dependency in GPS is in the number of backlogged connections. There is no further latitude and sharing is determined by the guaranteed rates which are set based on long term needs of the connections. This restriction on bandwidth sharing is more stringent than that necessary to preserve a key of property of fair queuing, the ability to guarantee worst case delay bounds for leaky bucket controlled traffic sources. Consequently, there is no need for fair queuing systems to emulate the possibly suboptimal excess bandwidth sharing of GPS.

It would thus be highly desirable to provide in a weighted fair queuing system emulating GPS, a method of achieving redistribution of unused bandwidth in a state-dependent manner, i.e., that reflects instantaneous needs of the remaining backlogged traffic flows.

Summary of the Invention

The instant invention is a modified approach to weighted fair queuing implementing an adaptive redistribution scheme. In such a scheme, each per connection flow is guaranteed its specified share of the link bandwidth with any excess bandwidth being adaptively redistributed. The scheme enables preservation of fair queuing's ability to provide worst case end to end delay bounds and the schemes work like fair queuing when there is no excess bandwidth. The excess bandwidth may be distributed according to a different criteria. Examples of state dependent criteria are: 1) Longest delay first (LDF) that serves the flow with current longest delay; 2) Least time to overflow (LTO) that serves the flow with minimum difference between maximum allowed delay and current delay; 3) Least time to overflow with leaky buckets (LTO-LB) that serves the flow which would cause buffer overflow first if worst case arrivals happen.

Advantageously, the adaptive bandwidth redistribution scheme always provides the minimum guarantees required for each of the connections and satisfies worst case fairness measures.

The LDF policy uses excess bandwidth to reduce the variance of the delay distribution which has the added benefit of reducing the playout buffer size for voice and video sources. Simulations with video traces and with voice traffic shows that indeed this policy performs better than GPS without any sacrifice of worst case guarantees. Since the deviation from the maximum allowed delay is not taken into account, flows with small delay bounds (like voice) get almost no excess bandwidth in the presence of flows with large delay bounds. Impreciseness in assigning weights can result in these flows experiencing losses much more than flows with large delay bounds.

The LTO policy tries to minimize packet losses by assigning excess bandwidth under the assumption that the flow which is likely to overflow the quickest has the most instantaneous bandwidth need. In doing so, it takes into account the current deviation of each flow from its maximum allowed delay. Simulations with a mix of CBR, voice, and video sources with very different delay bounds show that this policy reduces losses for all classes as well as reduces the variance of delay for each class. The various features of novelty which characterize the invention are pointed out

with particularity in the claims annexed to and forming a part of the disclosure. For a better understanding of the the invention, its operating advantages, and specific objects attained by its use, reference should be had to the drawings and descriptive matter in which there are illustrated and described preferred embodiments of the invention.

Brief Description of Drawings

FIG. 1 is a general data flow diagram illustrating the features of the weighted fair queuing and a rate proportional Server.

FIG. 2 illustrates the weighted fair queuing scheduling system of the invention.

Detailed Description of the Invention

Figure 2 is a general block diagram illustrating the fair queueing method with adaptive excess bandwidth redistribution. The system consists of a set of queues 100, a shaper device 30a,..30i associated with each queue, and Weighted Fair Queueing Server or Rate Proportional Server 40 and a State Dependent Server (SDS) 200. Packets are temporarily stored in the queues 20a,...20i. A weight is assigned with each of these queues representing the portion of the output bandwidth that must be allocated to the packets arriving in each of the queues. The shaper device is forwarding packets from the queues to the Weighted Fair Queueing Server with a rate exactly equal to the allocated. This scenario enables excess bandwidth to become available for redistribution. In accordance with the principles

of the invention, traffic from another source, i.e., another queue, may be directed to the adaptive bandwidth redistribution mechanism, which is hereafter referred to as state dependent scheduler 200. It should be understood to skilled artisans that the hardware architectures for implementation of the per-connection queues, shaper devices, and schedulers may vary, without diverting or straying from the principles of the invention described herein.

As described above, the shaper releases packets to the scheduler with a rate exactly equal to the allocated rate. The service offered by the shapers 30a,..,30i to the respective queue connections "*i*" during an interval of time is denoted by $S_i(\tau,t)$ in Figure 2. It is assumed that packets are transferred from the shaper to the scheduler with infinite capacity. The service offered by the rate proportional scheduler 40 is denoted as $R_i(\tau,t)$.

Packets that have not become eligible for service remain in the corresponding connection queue in the shaper, while all the eligible packets wait for service in the RPS Scheduler queue 40. Service is always provided from the RPS scheduler queues 40, as long as packets are available there. When all of the scheduler queues are empty, the state SDS 200 is invoked to select a packet from the queues 20a,..,20i for transmission with the service offered to a connection not affecting the state of the shaper 30. As shown in Fig. 2, the service offered by the shaper 30 to the queue connection "*i*" during an interval of time is denoted by $D_i(\tau,t)$ and the service offered by the state dependent scheduler 200 is denoted as $Z_i(\tau,t)$.

The shaper device 30 can be implemented using a calendar queue that is a well known mechanism to any skilled artisans. A description of the calendar queue mechanism for shaping has been presented in D. Stiliadis and A. Verma: A General Methodology for Designing Scheduling and Shaping Algorithms, in Proceedings of IEEE INFOCOM '97.

The Rate Proportional Server (Weighted Pair Queueing Server) 40 can be implemented using any known fair queueing mechanisms of which there are several known to those skilled in the art. Examples are D. Stiliadis and A. Varma: Traffic Scheduling System and Method for Packet-Switched Networks, U.S. Patent Application, Serial No, 08/634,904, filed April 15,1996.

At some point of time, if it is determined that the are no packets in the RPS scheduler, then this is an indication of free bandwidth. Thus, packets may be served by the SDS 200 based on a state dependent manner. By state dependent manner, it is meant that the decision is based on the current state of some variables related to the system. Two examples of state dependent decisions are Longest Delay First and Least Time to Overflow.

In the Longest Delay First, the scheduler SDS 200 selects among the queues 20a,...20i, the one that has the packet that will see the longest delay if this queue is served with a rate equal to guaranteed, To do that, the scheduler assigns to each queue a delay value that is equal to the size of the queue divided by the guaranteed rate. This value describes the delay that the last packet of this queue will see. The SDS 200 selects the queue with the maximum such value using any mechanism for selecting the maximum value among a set of numbers. The selection mechanism is a well known method to any skilled artisan.

In the Least Time to Overflow mechanism, the SDS scheduler 200 selects among the queues 20a,...,20i that queue which is likely to overflow in the shortest time. The mechanism assumes that a maximum size is associated with a queue. Every time a packet arrives or is serviced by a queue, a packet-to-overflow variable counts the number of packets that can be added to the queue without exceeding its maximum capacity. A second variable, denoted as time-to-overflow, indicates the expected time to overflow and is calculated by dividing the packet-to-overflow variable by the associated rate. The SDS 200 selects for transmission the queue with the minimum time-to-overflow variable. The minimum is determined by any of the well known mechanisms to anyone skilled in the art. Although the above variables explain two methods for calculating state

dependent variables, other similar methods may be used. Based on the state dependent variables, the scheduler mechanism 200 will direct packets from the selected queues over separate link 75 where the packet will be multiplexed by multiplexer element 350 and forwarded to its next destination.

It should be understood that the property of worst case fairness is also satisfied in the system of the invention. In worst case fairness, the servicing of queues of different lengths at different allocated rates is interleaved such that a packet from a first queue is serviced at time t_1 and the packet of a second queue is serviced at a time t_2. In the invention, the measure of worst ease fairness is satisfied, such that, the time interval t_2-t_1, in the worst case, is less than or bounded by a value that is not a function of the number of connections, but is a function of the packet size of the longest queue, e.g., the queue serviced at time t_1 and the allocated rate of that longest queue.

Thus, the property of worst case fairness is satisfied by the implementation of the SDS 200 in the weighted fair queuing system of the invention.

The foregoing merely illustrates the principles of the present invention. Those skilled in the art will be able to devise various modifications, which although not explicitly described or shown herein, embody the principles of the invention and are thus within its spirit and scope.

Claims

1. Apparatus for routing packets in a communication network comprising:

   a plurality of per-connection queues, each queue established for receiving packets from a

respective source and temporarily storing received packets before routing to a particular destination;

weighted fair queuing scheduling means servicing packets from each of said plurality of per-connection queues at guaranteed pre-allocated rates;

means for sensing a presence or absence of packets in queues, said absence of packets in queues indicating availability of excess bandwidth; and,

state dependent scheduling means for redistributing excess bandwidth upon sensing of queues absent said packets, said state dependent scheduling means servicing those queues in accordance with a state variable corresponding to a performance property of said queues,

wherein delay and isolation properties for routing packets of respective queues is preserved.

2. Apparatus for routing packets in a communication network as claimed in Claim 1, wherein a said performance property of a queue includes the queue having the greatest amount of packets, said corresponding state variable being longest queue first.

3. Apparatus for routing packets in a communication network as claimed in Claim 1, wherein a said performance property of a queue includes that queue having a packet waiting to be serviced in the longest amount of time, said corresponding state variable being longest delay first.

4. Apparatus for routing packets in a communication network as claimed in Claim 1, wherein a said performance property of a queue includes the queue most likely to overflow buffer memory containing said queue, said corresponding state variable being most likely to overflow buffer.

5. Apparatus for routing packets in a communication network as claimed in Claim 1, wherein a worst case fairness measure for said connections is satisfied.

6. Method for routing packets in a communication network comprising:

establishing a plurality of per-connection queues, each queue for receiving packets from a respective source and temporarily storing received packets before routing to a particular destination;

servicing packets from each of said plurality of per-connection queues at guaranteed pre-allocated rates;

sensing a presence or absence of packets in queues, said absence of packets in queues indicating availability of excess bandwidth; and,

for redistributing excess bandwidth upon sensing of queues absent said packets, said state dependent scheduling means servicing those queues in accordance with a state variable corresponding to a performance property of said queues, while preserving delay and isolation properties for routing of said packets.

FIG. 1

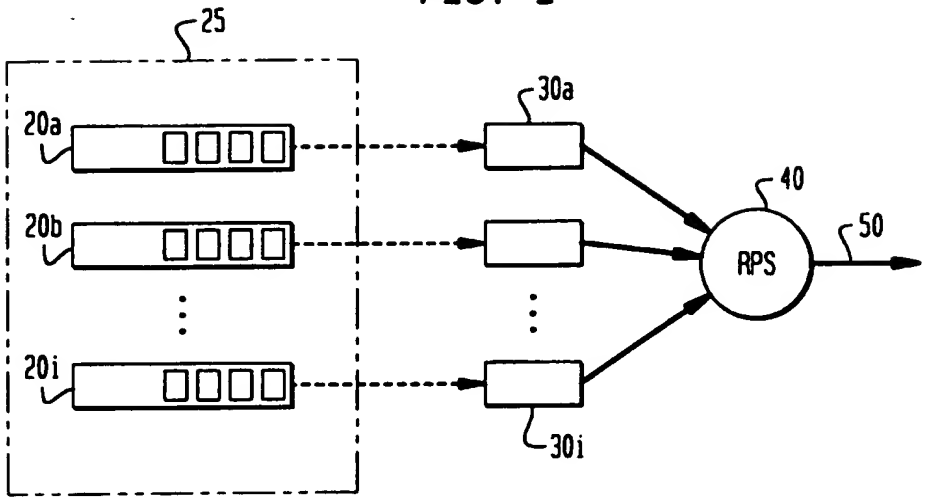
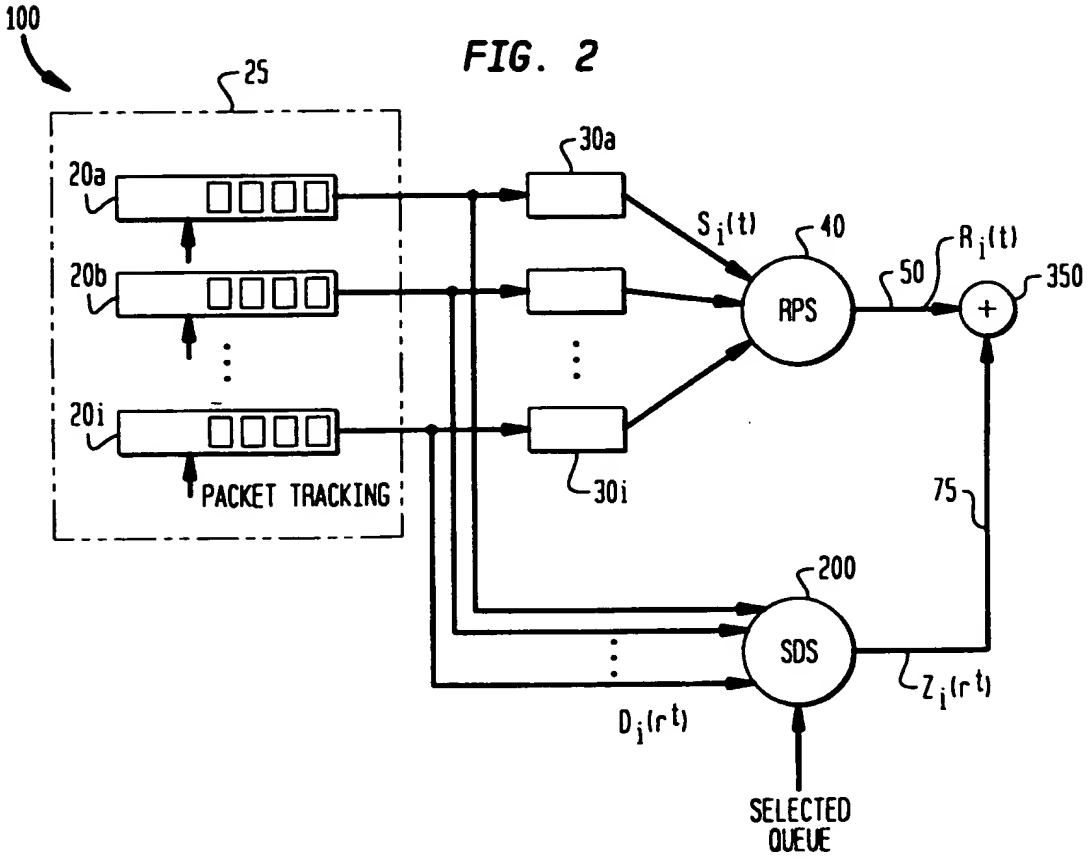

25
20a
20b
20i
30a
30i
40
RPS
50

FIG. 2


100
25
20a
20b
20i
PACKET TRACKING
30a
30i
$S_i(t)$
40
RPS
50
$R_i(t)$
350
+
75
200
SDS
$Z_i(r\,t)$
$D_i(r\,t)$
SELECTED QUEUE